

[19] 中华人民共和国国家知识产权局

[51] Int. Cl<sup>7</sup>



# [12] 发明专利申请公开说明书

[21] 申请号 02828899.8

G06F 15/00

G06K 1/00

H04N 1/40

H04N 1/46

B41J 29/38

B41J 2/205

B41J 2/045

[43] 公开日 2005 年 6 月 15 日

[11] 公开号 CN 1628294A

[22] 申请日 2002.3.25 [21] 申请号 02828899.8

[86] 国际申请 PCT/US2002/009155 2002.3.25

[87] 国际公布 WO2003/084051 英 2003.10.9

[85] 进入国家阶段日期 2004.11.5

[71] 申请人 时间工作室公司

地址 美国马萨诸塞州

[72] 发明人 阿达姆·L·卡雷

[74] 专利代理机构 中国国际贸易促进委员会专利商  
标事务所

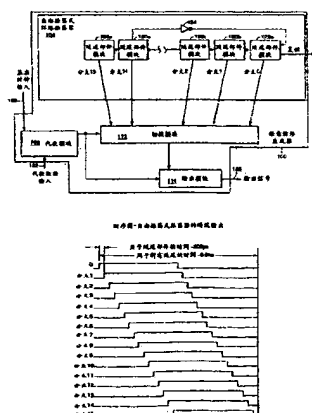
代理人 李 玲

权利要求书 5 页 说明书 32 页 附图 24 页

[54] 发明名称 使用自由振荡式环形振荡器的任意  
波形合成器

[57] 摘要

一种波形生成器，包括自由振荡式环形振荡器(104)、代数模块(108)、切换模块(112)以及输出模块(114)。自由振荡式环形振荡器包括多个连接在环路中的延迟部件(120p, 120o, 120c, 120b, 120a)以及排列在延迟部件之间的多个分支(分支15, 分支14, 分支2, 分支1, 分支0)，每个分支提供唯一定相的振荡跃迁信号。代数模块(108)对输入信号(132)作出响应，从而产生表示任意波形的第一上升沿的输出信号(136)。



ISSN 1008-4274

1. 一种用于提供任意波形的波形生成器，包括：

一个自由振荡式环形振荡器，包括：

多个连接在环路中的延迟部件；以及

排列在延迟部件之间的多个分支，每一个分支都提供了一个唯一定相的振荡跃迁信号；

一个代数模块，该模块具有一个代数数据输入端口、一个时钟输入端口以及一个代数数据输出端口，所述代数模块对在代数数据输入端口接收的信号作出响应，从而在代数数据输出端口产生一个表示任意波形的第一上升沿的信号；

一个切换模块，该模块具有一个与代数数据输出端口进行电通信的切换输入端口、多个与所述自由振荡式环形振荡器的分支进行电通信的切换分支输入端口以及一个切换输出端口，其中所述切换模块对在切换输入端口接收的表示第一上升沿的信号做出响应，从而在切换输出端口提供一个从多个分支的一个分支中选出的第一跃迁信号；以及

一个输出模块，该模块具有一个与切换输出端口进行电通信的跃迁信号输入端口、一个与代数数据输出端口进行电通信的窗口输入端口以及一个与代数模块的时钟输入端口进行电通信的波形输出端口，其中所述输出模块对在输出模块的跃迁信号输入端口接收的第一跃迁信号以及在窗口输入端口接收的表示第一上升沿的信号作出响应，从而在波形输出端口生成一个任意波形。

2. 权利要求 1 的任意波形生成器，还包括一个环路平均模块，该模块包括一个基准时钟输入端口和一个分支输入端口，其中所述分支输入端口电连接到多个分支，其中所述环路平均模块对在基准时钟输入端口接收的基准时钟信号作出响应，从而被配置成 (i) 对在基准时钟信号边沿之间出现的分支跃迁数目进行计数，(ii) 响应于所述计数对平均环路速度进行计算。

3. 权利要求 2 的环路平均模块, 还被配置成对 (i) 计算得到的平均环路速度以及 (ii) 捕捉到的多个环路状态作出响应, 从而计算环形振荡器的瞬时相位。

4. 权利要求 1 的任意波形生成器, 还包括:

一个微小延迟模块, 该模块包括一个与切换输出端口进行电通信的信号输入端口、一个与代数模块的代数数据输出端口进行电通信的选择输入端口以及一个与输出模块的输入端口进行电通信的微小延迟输出端口, 响应于在选择输入端口接收的指示第一上升沿的信号, 所述微小延迟模块对从切换模块的切换输出端口到输出模块的输入端口的第一跃迁信号的传播进行延迟;

其中所述输出模块的输入端口与切换模块的切换输出端口通过微小延迟模块进行电通信。

5. 权利要求 1 的任意波形生成器, 其中代数模块还包括一个同步输入端口, 响应于在代数数据输入端口接收的信号以及在同步输入端口接收的信号, 所述代数模块在代数数据输出端口生成一个表示任意波形内的第一上升沿的信号。

6. 权利要求 5 的任意波形生成器, 其中所述代数模块生成一个表示第一上升沿的信号, 由此在任意波形中实现第一上升沿的无缝合并。

7. 权利要求 1 的任意波形生成器, 其中一个延迟部件是反相延迟部件。

8. 权利要求 1 的任意波形生成器, 其中每一个延迟部件都是同等负载的。

9. 权利要求 1 的任意波形生成器, 其中至少有一个延迟部件还包含测试切换模块, 该模块具有一个测试控制输入端口和一个测试数据输入端口, 响应于在测试控制输入端口接收的信号, 所述延迟部件在相应分支生成一个与在测试数据输入端口接收的信号相同的信号。

10. 权利要求 1 的任意波形生成器, 还包括一个补偿模块, 该模块包括一个与代数模块进行电通信的数据输出端口以及一个用于对表

示基准时钟源的频率改变参数的信号进行接收的补偿输入端口，其中所述补偿模块对与频率改变参数相关联的基准时钟频率变化进行估计，响应于所述变化，所述补偿模块在补偿输出端口生成一个校正信号。

11. 权利要求 10 的任意波形生成器，其中频率改变参数是基准时钟源的温度。

12. 权利要求 10 的任意波形生成器，其中频率改变参数是基准时钟源的寿命。

13. 权利要求 1 的任意波形生成器，其中代数模块还被配置成对在代数输入端口接收的信号作出响应，从而在代数数据输出端口生成一个表示任意波形内的第一下降沿的第二信号。

14. 权利要求 1 的任意波形生成器，其中输出模块还被配置成生成一个捕获信号，该信号表示的是用于指示在窗口输入端口接收的第一上升沿的错误信号。

15. 权利要求 1 的任意波形生成器，其中所述任意波形生成器是在 CMOS 集成电路内部实现的。

16. 权利要求 1 的任意波形生成器，还包括：

一个放大器，所述放大器对来自耦合到环形振荡器的电源的电源噪声的至少一个分量进行前馈；以及

一个环路速度补偿电路，所述环路速度补偿电路与所述放大器进行电通信。

17. 权利要求 1 的任意波形生成器，还包括：

多个电容部件，其中每一个部件都与多条信号路径中的一条相应路径进行电通信，每一个电容部件都具有相应的预定容量，以便在经过相应信号路径传播跃迁信号的过程中生成相应的预定时延，由此，第一信号路径上的跃迁的传播时间实质上与第二信号路径上的跃迁的传播时间相等。

18. 权利要求 17 的任意波形生成器，其中多个电容部件中的每一个电容部件都具有小于 50 毫微微法左右的值。

19. 权利要求 17 的任意波形生成器, 其中多条信号路径中的每一条路径都与第一电容部件和第二电容部件进行电通信。

20. 权利要求 19 的任意波形生成器, 其中相对于温度而言, 第一电容部件和第二电容部件具有不同的灵敏度。

21. 权利要求 19 的任意波形生成器, 其中相对于工艺而言, 第一电容部件和第二电容部件具有不同的灵敏度。

22. 权利要求 19 的任意波形生成器, 其中相对于供应电压而言, 第一电容部件和第二电容部件具有不同的灵敏度。

23. 一种用于产生任意波形的方法, 包括:

在延迟部件环路中生成一个连续的跃迁序列;

对在环路中的一个延迟部件上出现的跃迁进行计数;

确定所需的第一上升沿时间和第一下降沿时间, 以便产生任意波形;

基于计数得到的跃迁和内部时钟对所需的第一上升沿时间做出响应, 以便选择连续跃迁序列中的第一跃迁;

基于计数得到的跃迁和内部时钟对所需的第一下降沿时间作出响应, 以便选择连续跃迁序列中的第二跃迁;

使用第一和第二选定跃迁来产生一个输出信号, 从而生成任意波形; 以及

基于输出信号生成内部时钟信号。

24. 权利要求 23 的方法, 其中生成输出信号的步骤还包括对过去出现的所需第一上升沿时间作出响应, 从而使用连续跃迁序列中的每一个跃迁来产生输出信号。

25. 权利要求 23 的方法, 还包括相对于输入基准时钟而在其中一个延迟部件上计算平均跃迁速率。

26. 权利要求 25 的方法, 还包括对 (i) 计算得到的平均跃迁速率以及 (ii) 延迟部件环路的多个捕捉状态作出响应, 从而计算延迟部件环路内的瞬时相位。

27. 权利要求 23 的方法, 还包括将多个传播延迟中的一个延迟

添加到选定跃迁中，其中在多个传播延迟中的两个延迟之间的差值小于跃迁经由延迟部件环路中的延迟部件的平均传播时间。

28. 权利要求 23 的方法，还包括：

接收一个外部同步信号；以及

对接收到的同步信号作出响应，从而修改第一跃迁和第二跃迁的选择。

29. 权利要求 28 的方法，其中修改步骤还包括修改第一跃迁的选择，由此在输出信号中提供第一上升沿的无缝合并，以便生成任意波形。

30. 权利要求 23 的方法，其中每一个跃迁都具有极性，并且生成连续跃迁序列的步骤还包括使用一个延迟部件来对跃迁极性进行反相。

31. 权利要求 23 的方法，其中生成连续跃迁序列的步骤还包括为每个延迟部件提供相等的负载。

32. 权利要求 23 的方法，还包括：

接收一个表示基准时钟源的频率改变参数的信号；

确定频率改变参数所造成的基准时钟变化；以及

对所确定的变化作出响应，从而改变第一跃迁的选择。

33. 权利要求 32 的方法，其中频率改变参数是基准时钟源的温度。

34. 权利要求 32 的方法，其中频率改变参数是基准时钟源的寿命。

35. 权利要求 23 的方法，还包括将预定时延添加到经由相应信号路径的跃迁传播之中，由此，经过信号路径的跃迁的传播时间实质上与经过第二信号路径的跃迁的传播时间相等。

## 使用自由振荡式环形 振荡器的任意波形合成器

### 技术领域

本发明一般涉及波形生成领域。更准确地说，本发明涉及一种通过使用自由振荡式环形振荡器来产生任意波形的方法和设备。

### 背景技术

目前存在很多希望将时间单位或距离单位以廉价方式分成更小单位的情况。例如在一个范围检测器中，输出信号与输入信号之间所经过的时间可能短到只有若干毫秒，使用标准的基准时钟来计算流逝的时间将会产生错误计数，由此导致不准确的距离计算。

在激光打印机这样的另一个实例中，较为理想的是在用于高分辨率的给定行单元内部的不同距离提供一个从白到黑的跃迁（反之亦然）。相关内容可以参见在此引入作为参考的美国专利 5,109,283。然而，在打印机的激光二极管上并不足以分辨那些以 10MHz 的打印机像素时钟速率来计时的黑色-白色信号或白色-黑色信号，由此不能对指定打印单元内部的打印跃迁执行精确控制。更快的时钟则过于昂贵和/或不能用于在高分辨率图形图像所需要的高时间分辨率上为打印跃迁提供基准（例如与微米级的打印单元距离相对应的 1 纳秒）。

一种解决上述问题的方案是使用一个结合了自由振荡式（free-running）环形振荡器（也可以将其称为环路振荡器）的设备。在美国专利 5,793,709 和 5,903,522 中描述了这种解决方案，这些专利在此引入作为参考。然而，对 CD-RW、DVD、通信和雷达领域中的应用来说，其在信号方面的需求要求使用具有更高频率和精度的跃迁边沿，由此对其产生了限制。而电路传导路径和实施硬件中的差别也会变得非常明显。基于基准时钟频率的代数时钟将会限制代数电路执

行计算的速度。此外，计算自由振荡式环形振荡器（“环路”）的外部电路同样限制了代数电路执行计算的速度。在使用同步信号时，使用所述输出的设备无法容忍输出信号中的不可预测的突然跃迁。

### 发明内容

本发明旨在解决这些需要。

因此，本发明的一个目的是对不同实施硬件和不等路径长度进行补偿。本发明的另一个目的是根据生成的输出信号来产生一个远比基准时钟更快的用于跃迁边沿计算的内部时钟。本发明还包括一个用于在跃迁边沿计算出错时提供内部时钟的监视器（watchdog）。此外，本发明的另一个目的是通过在内部计算平均环路速度来提高总的可允许的计算频率。所述电路还根据平均环路速度而对环路中的跃迁位置进行计算，从而提高精度。本发明的另一个目的是在输出相位发生预定变化时产生一个将更新输出波形与先前输出波形无缝结合的输出。对 CD-RW、DVD、通信和雷达领域中的应用来说，它们需要的是更快和更准确的边沿跃迁。

本发明涉及一种用于产生任意波形的方法和设备，在一个方面中，本发明涉及一种用于产生任意波形的波形生成器。所述波形生成器包括一个自由振荡式环形振荡器、一个代数模块、一个切换模块以及一个输出模块。自由振荡式环形振荡器包括多个连接在环路中的延迟部件以及排列在延迟部件之间的多个分支，其中每一个分支都提供了一个唯一定相的振荡跃迁信号。代数模块则包括一个代数数据输入端口、一个时钟输入端口以及一个代数数据输出端口。所述代数模块对在代数数据输入端口接收的信号作出响应，从而在代数数据输出端口产生一个表示任意波形的第一上升沿的信号。切换模块则包括一个与代数数据输出端口进行电通信的切换输入端口、多个与自由振荡式环形振荡器的分支进行电通信的切换分支输入端口以及切换输出端口。在切换输出端口，切换模块对在切换输入端口接收的表示第一上升沿的信号做出响应，从而提供一个从多个自由振荡式环形振荡器分



支的一个分支中选出的第一跃迁信号。输出模块则包括一个与切换输出端口进行电通信的跃迁信号输入端口、一个与代数数据输出端口进行电通信的窗口输入端口以及一个与代数模块的时钟输入端口进行电通信的波形输出端口。所述输出模块还对在输出模块的跃迁信号输入端口接收的第一跃迁信号以及在窗口输入端口接收的第一上升沿的信号作出响应，从而在波形输出端口创建一个任意波形。

在一个实施例中，任意波形生成器包括一个环路平均模块，所述环路平均模块则包括一个基准时钟输入端口和一个分支输入端口，所述分支输入端口电连接到自由振荡式环形振荡器中的多个分支。所述环路平均模块对在基准时钟输入端口接收的基准时钟信号作出响应，从而对在基准时钟信号边沿之间出现的分支跃迁数目进行计数，此外还响应于所述计数对平均环路速度进行计算。所述环路平均模块还可以对计算得到的平均环路速度以及捕捉到的多个环路状态作出响应，从而计算环形振荡器的瞬时相位。

在另一个实施例中，任意波形生成器包括一个微小延迟模块。所述微小延迟模块包括一个与切换输出端口进行电通信的信号输入端口，一个与代数模块的代数数据输出端口进行电通信的选择输入端口，以及一个与输出模块的输入端口进行电通信的微小延迟输出端口。响应于在选择输入端口接收的第一上升沿的信号，所述微小延迟模块对从切换模块的切换输出端口到输出模块的输入端口的第一跃迁信号的传播进行延迟。此外，所述输出模块的选择输入端口还通过微小延迟模块与切换模块的切换输出端口进行电通信。

在另一个实施例中，代数模块包括一个同步输入端口。响应于在代数数据输入端口和同步输入端口接收的信号，所述代数模块在代数数据输出端口创建一个处于任意波形内部的第一上升沿的信号。然后，所述代数模块产生一个第一上升沿的信号，由此在任意波形生成器中实现第一上升沿的无缝合并。

在另一个实施例中，其中一个延迟部件是反相延迟部件，并且每个延迟部件都是同等负载的。在另一个实施例，至少有一个延迟部件

包含了测试切换模块。这个测试切换模块具有一个测试控制输入端口和一个测试数据输入端口。响应于在测试控制输入端口接收的信号，所述延迟部件会在相应分支创建一个与在测试数据输入端口接收的信号相一致的信号。

在另一个实施例中，任意波形生成器还包括一个补偿模块。所述补偿模块包括一个与代数模块进行电通信的数据输出端口，以及一个用于对表示基准时钟源的频率改变参数的信号进行接收的补偿输入端口。所述补偿模块对关联于基准时钟源的频率改变参数的基准时钟频率变化进行估计。然后，响应于这个频率变化，所述补偿模块在补偿输出端口创建一个校正信号。在另一个实施例中，频率改变参数是基准时钟源的温度。而在另一个实施例中，频率改变参数是基准时钟源的寿命。

在另一个实施例中，代数模块对在代数输入端口接收的信号作出响应，从而在代数数据输出端口创建任意波形内部的第一下降沿的第二信号。在另一个实施例中则使得输出模块创建一个关于在窗口输入端口接收的第一上升沿的错误信号的追逐信号。在另一个实施例中，所述任意波形生成器是在 CMOS 集成电路内部实现的。

在另一个实施例中，任意波形生成器包括一个放大器和一个环路速度补偿电路。放大器对来自耦合到环形振荡器的电源的电源噪声的至少一个分量进行前馈，环路速度补偿电路则与放大器进行电通信。在一个实施例中，响应于经过放大的噪声，环路速度补偿电路与环路平均模块进行电通信，以便对所计算的平均环路速度进行调整。在另一个实施例中，环路速度补偿电路与代数模块进行电通信，以便响应于经过放大的噪声对计算得到的平均环路速度进行调整。

在另一个实施例中，任意波形生成器包括多个电容部件，其中每一个部件都与多条信号路径中的一条相应路径进行电通信。每一个电容部件都具有相应的预定容量，以便在经由相应信号路径传播跃迁信号的过程中创建相应的预定时延，这样一来，第一信号路径上的跃迁的传播时间实质上与第二信号路径上的跃迁的传播时间是相等的。

在另一个实施例中，多个电容部件中的每一个电容部件都具有一个小于 50 毫微微法左右的值。在另一个实施例中，多条信号路径中的每一条路径都与第一电容部件和第二电容部件进行电通信。并且在另一个实施例中，相对于温度、工艺以及电源电压而言，第一和第二电容部件具有不同的灵敏度。

在另一个方面中，本发明涉及一种用于创建任意波形的方法。所述创建任意波形的方法包括在延迟部件环路中生成一个连续跃迁序列，对在环路中的其中一个延迟部件上出现的跃迁进行计数，确定所需的第一上升沿时间和第一下降沿时间，基于计数得到的跃迁和内部时钟对所需第一上升沿做出响应，以便选择连续跃迁序列中的第一跃迁，基于计数得到的跃迁和内部时钟对所需第一下降沿时间作出响应，以便选择连续跃迁序列中的第二跃迁，通过使用第一和第二选定跃迁来产生一个输出信号，以及基于输出信号生成内部时钟信号。

在一个实施例中，生成输出信号的方法包括对过去出现的所需第一上升沿时间作出响应，从而使用连续跃迁序列中的每一个跃迁来产生输出信号。

在另一个实施例中，所述方法包括相对于输入基准时钟而在其中一个延迟部件上计算平均跃迁速率。

在另一个实施例中，所述方法包括对计算得到的平均跃迁速率以及延迟部件环路的多个所捕捉状态作出响应，从而计算延迟部件环路内部的瞬时相位。

在另一个实施例中，所述方法包括将多个传播延迟中的一个延迟添加到选定跃迁中，其中在多个传播延迟中，两个延迟之间的差值小于一个跃迁经由延迟部件环路中的延迟部件的平均传播时间。

在另一个实施例中，所述方法包括接收一个外部同步信号，以及对接收到的同步信号作出响应，从而修改第一和第二跃迁的选择。在另一个实施例中，所述修改处理包括调整第一跃迁的选择，由此在输出信号中提供第一上升沿的无缝合并，从而创建任意波形。

在另一个实施例中，每一个跃迁都具有极性。并且在另一个实施

例中，产生连续跃迁序列包括结合一个延迟部件来对跃迁极性进行反相。此外在另一个实施例中，连续跃迁序列的生成为各个延迟部件都提供了相等的负载。

在另一个实施例中，所述方法包括接收一个表示基准时钟源的频率改变参数的信号，确定频率改变所造成的基准时钟的任何变化，以及对所确定的变化作出响应，从而改变第一跃迁的选择。并且在另一个实施例中，所述频率改变参数是基准时钟源的温度。此外，在另一个实施例中，所述频率改变参数是基准时钟源的寿命。

在另一个实施例中，所述方法包括将预定时延添加到经由相应信号路径的跃迁传播之中，由此经过信号路径的跃迁的传播时间实质上与经过第二信号路径的跃迁的传播时间是相等的。

#### 附图说明

本领域技术人员可以从以下关于优选实施例的描述以及附图中想到其他目标、特征和优点，其中：

图 1A 是本发明一个实施例的高级框图；

图 1B 是来自图 1A 所示的自由振荡式环路振荡器分支的信号的时序图；

图 2 是描述附加模块的本发明的一个实施例的高级框图；

图 3 是图 2 所示的延迟部件模块的一个实施例的详细框图；

图 4 是本发明另一个实施例的高级框图；

图 5 是本发明的代数模块的一个实施例的详细框图；

图 6 是图 5 所示的周期积分器模块的一个实施例的详细框图；

图 7 是图 5 所示的环路平均模块的一个实施例的详细框图；

图 8 是图 5 所示的时钟域变换模块的一个实施例的详细框图；

图 9A 是图 5 所示的同步模块的一个实施例的详细框图；

图 9B 是图 9A 所示的同步去抖动模块的一个实施例的详细框图；

图 10A 是本发明的切换模块的一个实施例的详细框图；

图 10B 是图 10A 所示的平衡复用模块的一个实施例的详细框图；

图 11 是本发明的微小延迟模块的一个实施例的详细框图；

图 12 是本发明的输出模块的一个实施例的详细框图；

图 13 是图 12 所示的捕获模块、窗口模块和边沿触发器的一个实施例的详细框图；

图 14 是图 12 所示的相位组合模块的一个实施例的详细框图；

图 15 是图 14 所示的平衡 XOR 模块的一个实施例的详细框图；  
以及

图 16 是通过本发明一个实施例所进行的信号传播的时序图。

### 具体实施方式

图 1A 在高级别上描述了一个用于创建输出信号 136 的任意波形生成器 100。所述任意波形生成器 100 包括一个自由振荡式环路振荡器 104、一个切换模块 112、一个代数模块 108 以及一个输出模块 114。环路振荡器 104 经由分支电连接到切换模块 112，以便将跃迁边沿传送到切换模块 112。代数模块 108 接收一个外部基准时钟信号 128 和一个代数数据输入信号 132。如下文更详细描述的那样，代数模块 108 通过使用这些信号 128、132 而从分支中选择跃迁边沿，以便产生输出信号 136 的下一个跃迁边沿。代数模块 108 电连接到切换模块 112 和输出模块 114，以便传送计算得到的用于产生输出信号 136 的跃迁边沿。切换模块 112 接收来自代数模块 108 的跃迁边沿信息，并且选择与计算得到的跃迁边沿相对应的环路振荡器 104 的分支。所述切换模块还电连接到输出模块 114，以便将来自选定分支的跃迁边沿传送到输出模块 114。输出模块则接收跃迁边沿，并且将其作为输出信号 136 来进行供应。

自由振荡式环路振荡器 104 包括一个针对外部复位信号 140 的输入端、一系列延迟部件模块 120a、120b、120c、120o、120p（概括为 120）以及至少一个反相部件 124。复位信号 140 具有两种状态，即复位状态和运行状态。当信号 140 处于复位状态时，环路振荡器 104 并不振荡。当信号 140 跃迁到运行状态时，环路振荡器 104 将会开始振

荡。所述跃迁则经由第一延迟部件模块 120a 传播到下一个延迟部件模块 120b 的输入端。在延迟部件模块 120 的输入端接收到信号之后不久，在相应分支上将会输出所述信号。

举例来说，在将跃迁传播到延迟部件模块 120a 的输入端之后的某个时间，所述跃迁将会出现在分支 0 上。由于所述跃迁还传播到延迟部件模块 120a 的输出端并且随后传播到下一个延迟部件模块 120b 的输入端，因此该跃迁还会在分支 1 上出现。这个处理会在所述跃迁前进到最后一个延迟部件模块 120p 和分支 15 的整个过程中重复。图 1A 的示范性实施例描述了 16 个分支。但是分支数目也可以改变。此外，分支数目是出于设计考虑而选择的，其数目部分地取决于所需频率、可用实施硬件以及代数模块计算速度。

图 1B 的示范性实施例描述的是在图 1A 的相应分支上出现的跃迁。如图 1B 所示，在各个分支上的各个跃迁之间存在着大约 300 皮秒的延迟。这个延迟时间表示的是所述跃迁从延迟部件模块 120 的输入端传播到相应分支的输出端所花费的时间。它取决于实现所述设计的硬件。而所示实施例则是在采用了 0.35 微米工艺的 ASIC 上实现的。采用 0.25 微米工艺的 ASIC 可以产生大小约为采用 0.35 微米工艺的 ASIC 所产生的延迟时间的一半的延迟时间。借助于更小的延迟时间，代数模块 108 可以计算出所述跃迁边沿在输出信号 136 中的更准确的位置。所述延迟时间则是出于设计考虑而选择的，它取决于所需频率、可用实施硬件以及代数模块计算速度。

在另一个实施例中，每一个延迟部件模块 120 都包含一个反相部件。所述反相部件将会使间隔分支在与围绕环路的单个边沿传播相反的方向上进行跃迁。例如，分支 1、分支 3、分支 5、分支 7、分支 9、分支 11、分支 13 以及分支 15 各自从高电平状态跃迁到低电平状态，而不像图 1B 所显示的那样从低电平状态跃迁到高电平状态。由于正跃迁的传播时间不同于负跃迁的传播时间，因此使用反相部件将会使围绕环路的传播时间更为一致。此外，反相部件还产生了更均匀的电

源负载。

参考图 2, 跃迁是通过一系列延迟部件模块 120a ~ 120p 传播的。一旦到达最后一个延迟部件模块 120p 的输出端, 那么所述跃迁将会完成一个循环 (也称为一个行程 (lap))。环路 108 的示范性实施例包括与最后一个延迟部件模块 120p 的输出端相连的反相逻辑部件 208。这个反相逻辑部件 208 与连接到在先延迟部件模块 120 的输出端的每一个延迟部件模块 120 中所包含的反相逻辑部件 404 (图 3) 相类似。由此可以确保最后一个延迟部件模块 120p 的输出是以类似于其他延迟部件模块 120a ~ 120o 的方式加载的, 这样一来, 在各个分支上, 跃迁之间的时间都是相似的。此外, 硬件实施方式的容限还可以确定延迟时间等同到何种程度。

为了创建“循环”效果, 在这里必须重复执行经由一系列延迟部件模块 120a、120b、120c、120o、120p 的跃迁传播。为此目的, 在这里选择一个跃迁 (例如从延迟部件模块 120o 开始的跃迁) 并且将其经由反相部件 124 回送到第一延迟部件模块 120a 的输入端。所使用的反相部件 124 的数目取决于选定延迟部件模块 120 (例如延迟部件模块 120o) 是否在第一延迟部件模块 120a 的相反方向上输出一个跃迁。

举例来说, 如果第一延迟部件模块 120a 先前是从低电平跃迁到高电平, 并且选定跃迁 (例如延迟部件模块 120o) 也是从低电平跃迁到高电平, 那么在所述路径中必然存在一个使得传送到第一延迟部件模块 120a 的输入端的跃迁是从高到低的反相器 124。如果所述实施例具有包含反相部件的延迟部件模块 120, 并且已经相对于第一延迟部件模块 120a 而对选定跃迁进行了反相, 那么这两个反相部件 124, 124' (未显示) 将会串联使用, 以便确保将相反方向的跃迁传送到第一延迟部件 120a。此外在这里还对连接到反相器 124 的延迟部件模块 120o 进行选择, 以使所述跃迁从选定延迟部件模块 120o 经由一个或多个反相部件 124 到达第一延迟部件模块 120a 的传播时间近似等于从选定延迟部件模块 120o 跃迁到最后一个延迟部件模块 120p 的输出端的传播时间。

对一个说明性实例来说, 每一个延迟部件模块 120 都包含了反相

部件。经过两个反相部件 124、124'（未显示）的传播时间与两个延迟部件模块 120 的传播时间是相等的。此外在这里还选择了对应于分支 13 的延迟部件模块 120n（未显示）。所述延迟部件模块 120n 是第一延迟部件模块 120a 的相反方向的跃迁，这样则需要两个反相部件 124、124' 来确保正确的极性。此外，所述定时同样是正确的，由此在该跃迁还在从分支 13 传播到分支 15 的时间里经由两个反相部件 124、124' 到达第一延迟部件模块 120a 的输入端。

如图 2 所示，环路振荡器 104 还包括一个行程计数器模块 200 和一个状态捕捉模块 204。代数模块 108 还包括一个环路平均模块 212。行程计数器模块 200 电连接到第一延迟部件模块 120a，以便在跃迁边沿经过延迟部件模块 120a 时接收所述跃迁边沿。行程计数器模块 200 电连接到状态捕捉模块 204，以便传送所保存的行程数目。状态捕捉模块 204 则接收一个基准时钟信号 128。所述状态捕捉模块 204 电连接到每一个分支以及行程计数器模块 200，以便在基准时钟信号 128 跃迁时接收并保存环路振荡器 104 的分支和行程计数器模块 200 的状态。此外，所述状态捕捉模块 204 还电连接到代数模块 108，以便传送所保存的状态信息 220。而环路平均模块 212 则接收来自状态捕捉模块 204 的经发送的存储的状态信息 220，并对在代数模块 108 的计算中所使用的平均环路速度进行计算。

行程计数器 200 包括两个计数器。其中正沿计数器 200a（未显示）在正沿跃迁（例如从逻辑低到高）每次经过第一延迟部件模块 120a 时都会计数 1。负沿计数器 200b（未显示）则会在负沿跃迁（例如从逻辑高到低的跃迁）每次经过第一延迟部件模块 120a 时计数 1。因此，负沿计数器 200b 与正沿计数器 200a 具有 180 度的相位差。在这里使用两个计数器 200a、200b 是为了确保即使在更新一个计数器时，所述任意波形生成器 100 的其他模块也可以使用稳定和精确的计数器来确定行程数目。每一个计数器的大小（位数）都是可以改变的，并且所述大小是出于设计考虑而选择的，它取决于所需频率、可用实施硬件以及代数模块计算速度。此外，每一个计数器都会保持计数，一旦达



到最大值, 则计数器回滚到零并且继续计数。

状态捕捉模块 204 包括一个将基准时钟信号 128 用作时钟的寄存器 204a (未显示)。一旦基准时钟信号 128 跃迁 (通常是正沿跃迁), 则寄存器 204a 保存 (也称为捕捉) 环路振荡器 104 的状态。在一个实施例中, 寄存器 204a 保存的是一个 16 位数据字。在本实施例中则将 11 个比特用于行程计数器 200 的值。此外还使用了四个比特来表示 16 个分支。在这里将这四个比特用于分支状态, 由此表示所述跃迁在捕捉时传播所经过的最后一个分支。此外在这里还使用了一个比特来表示所述跃迁在经过第一延迟部件模块 120a 时的跃迁状态 (也就是正沿或负沿)。为了确定代数模块 108 选择的跃迁边沿究竟是正沿还是负沿以及是否需要由切换模块 112 进行反相, 在这里需要使用这个比特。

延迟部件模块 120 创建那些输出到分支并被捕捉的跃迁边沿。图 3 中更详细地显示了其中一个延迟部件模块 120。延迟部件模块 120 包括一个 NAND 部件 404。NAND 部件 404 的两个输入端 408、412 分别接收由在先延迟部件模块 120 输出的信号以及一个逻辑高电平信号 424。所述逻辑高电平信号确保 NAND 部件 404 的输出即为来自先前延迟部件模块 120 的输出端的输入信号 408 的反相信号。

对第一延迟部件模块 120a (图 2) 来说, 逻辑高电平信号由复位信号 140 取代, 输入信号 408 则是从反相输出部件 124 而不是先前延迟部件 120 输出的。当复位信号 140 处于复位状态即逻辑低电平时, 无论输入信号 408 怎样变化, 所述 NAND 部件 404 输出的都是一个逻辑高电平。对自由振荡式环形振荡器 104' 来说, 其剩余延迟部件模块 120b 到 120p (图 2) 都会驱动到其相应状态, 所述环路振荡器 104 则会在复位信号跃迁到运行状态即逻辑高电平之前一直处于复位状态。

NAND 部件 404 的输出发送到下一个延迟部件模块 120。所述输出 412 同时也是反相器 416 的一个输入。来自某个选定延迟部件模块 120 (例如图 2 的延迟部件模块 120o) 的反相输出端 420 的跃迁发送到另一个反相器 124 (图 2), 然后则发送到第一延迟部件模块 120a, 从而保持环路振荡器 104 的振荡。对其他延迟部件模块 120a、120b、

120c、120p 来说，输出 420 并未连接。在一个实施例中，出于负载均衡目的，无论是否连接，每一个延迟部件模块 120 中都会包含反相器 416。如上所述，通过尽可能保持延迟部件模块 120 中的负载相似和相等，可以使得经过每一个延迟部件模块 120 的传播时间保持一致。

NAND 部件 404 的输出端 412 与测试切换模块 400 相连。所述测试切换模块 400 接收一个测试输入信号 440。这个信号 440 的状态确定了由测试切换模块 400 生成的信号。举例来说，如果测试输入信号 440 处于一种“NOT(否)”测试状态(例如逻辑低电平)，那么 NAND 部件 428 将会输出一个信号，该信号与 NAND 部件 404 的输出 412 具有相反的状态。无论测试数据输入信号 444 怎样变化，所述 NAND 部件 432 都会输出一个逻辑高电平。因此，NAND 部件 436 输出的是与 NAND 部件 428 的输出相反的状态，所述 NAND 部件 428 的输出则与 NAND 部件 404 的输出 412 的状态相同。如果测试输入信号 440 处于测试状态(例如逻辑高电平)，那么无论 NAND 部件 404 的输出 412 怎样变化，所述 NAND 部件 428 都会输出一个逻辑高电平。NAND 部件 432 输出的则是与测试数据输入信号 444 相反的状态。因此，NAND 部件 436 输出的是与 NAND 部件 432 的输出相反的状态，这个状态与测试数据输入信号 444 的状态则是相同的。

反相驱动器 448 对 NAND 部件 436 的输出进行放大，其中所述输出即为测试切换模块 400 的输出。反相驱动器 448 的输出则是发送到切换模块 112(图 2)的分支信号 456。因此，依照测试输入信号 440 的状态，反相驱动器 448 的输出将会是 NAND 部件 404 的输出或是测试数据输入信号 444 的相反状态。

对第一延迟部件模块 120a 来说，NAND 部件 436 的输出是反相驱动器 452 的一个输入。反相驱动器 452 的输出端与行程计数器模块 200 相连。行程计数器对从反相驱动器 452 输出的每一个跃迁进行计数。对其他延迟部件模块 120b 到 120p 来说，反相驱动器 452 的输出并未连接。在一个实施例中，为了均衡负载，无论是否连接，每一个延迟部件模块 120 中都会包含反相驱动器 452。

图4描述的是任意波形生成器100'的一个替换实施例。任意波形生成器100'还包括一个微小延迟模块310、一个补偿模块320、一个自由振荡式环路振荡器104'，其中所述振荡器包括一个状态捕捉模块204'和一个代数模块108'。所述代数模块108'则包括一个同步模块304。微小延迟模块310电连接到切换模块112，以便接收跃迁边沿。所述微小延迟模块310还经由一个延迟部件模块120而将差分延迟添加给小于传播延迟时间的跃迁边沿。此外，微小延迟模块310还电连接到输出模块114，以便将经过进一步延迟的跃迁边沿传送到输出模块114。补偿模块320则接收补偿输入信号324。此外，所述补偿模块还电连接到代数模块108'，以便传送调整信号。

在一个实施例中，补偿模块320接收一个补偿输入信号324，该信号表示的是创建基准时钟输入信号128的振荡器（也就是基准时钟128的来源）的温度。补偿模块320根据补偿输入信号324并且使用温度-频率的预定关系（例如查找表）来确定基准时钟128的频率变化。此外，补偿模块320还将带有变化信息的调整信号输出到代数模块108'。依照基准时钟128的周期增量的变化以代数方式添加给所需输出周期132，从而在没有恒温箱和加热延迟的情况下产生一个极其稳定的时间基准，以此作为输出信号136。

在另一个实施例中，通过将寿命标识信号（例如当前日期）用作补偿输入信号324，可以采用相似方式来对晶体老化进行校正。而补偿模块320则是根据补偿输入信号324并且使用寿命-频率的预定关系（例如查找表）来确定基准时钟128的频率变化的。

状态捕捉模块204'接收一个同步输入信号300。并且状态捕捉模块204'电连接到每一分支以及行程计数器模块200，以便在同步信号300变换时接收和保存环路振荡器104'分支以及行程计数器模块200的状态。所述状态捕捉模块204'包括一个将同步信号300用作时钟的第二寄存器204b（未显示）。一旦同步信号300跃迁（通常是正沿跃迁），那么第二寄存器204b将会保存环路振荡器104'的状态。在一个实施例中，第二寄存器204b保存一个16比特的数据字，它是用于

描述第一寄存器 204a 的比特。

状态捕捉模块 204'还电连接到代数模块 108',以便传送所保存的状态信息 330。代数模块 108'的同步模块 304 则接收状态捕捉模块 204'传送的被保存状态信息 330 并且计算后续的跃迁边沿选定范围,由此输出信号 136 与同步信号 300 处于相同相位。

图 5 更详细地描述了代数模块 108'。代数模块 108'对从环路振荡器 104'中创建输出信号 136 的每一个跃迁边沿所需要的行程和分支进行计算。在一个实施例中,四信道代数模块 108'计算输出信号 136 的接下来的两个上升沿以及对应的下降沿。第一上升沿称为相位 0 信号,对应的下降沿则称为相位 1 信号,并且第二上升沿称为相位 2 信号,而相应的下降沿则称为相位 3 信号。为使附图保持清楚,除非所指内容之外,在这里只对用于创建相位 0 信号的电路进行描述。此外,除非另有所指,否则用于其他三个相位信号的电路都是相似的。通过并行计算四个边沿,代数模块 108'可以在一个更快的频率上工作,由此允许输出信号 136 是一个更高的频率。在其他实施例中,根据所需要的性能而需要用到多于或少于四个信道。

代数模块 108'接收基准时钟 128 和 RISE(上升)时钟 540。RISE 时钟 540 是从输出模块 114 的相位组合模块 1225(图 12)那里接收的。所述 RISE 时钟 540 是输出信号 136 的频率的一半,它具有与输出信号 136 的上升沿相对应的边沿。代数模块 108'则是将 RISE 时钟 540 用作执行计算和将数据存入寄存器的时钟。此外,代数模块 108'还接收一个代数数据输入信号 132。所述代数数据输入信号 132 包括一个表示基准时钟输入信号 128 的周期乘数的数据字 132a,其乘积则限定了输出信号 136 的所需周期。

在一个实施例中,数据字 132a 是一个 27 比特的字,其中前 6 个比特表示乘数的整数部分,另外 21 个比特表示乘数的小数部分。使用 27 比特使得用户能够定义精度超过 1ppm 的输出信号 136 的频率。例如,对作为基准时钟输入 128 的周期的二分之一(也就是频率的两倍)的输出信号 136 来说,所述数据字 132a 是

000000.10000000000000000000。对作为基准时钟输入 128 的周期的四分之一(也就是频率的四倍)的输出信号 136 来说,所述数据字 132a 是 000000.01000000000000000000。在另一个实例中,作为周期为基准时钟输入 128 的周期的 2.25 倍的输出信号 136 (也就是频率的 0.444444..... 倍) , 所述数据字 132a 是 000010.01000000000000000000。

周期积分器模块 500 使用乘数数据字 132a 来确定何时出现下一个涉及基准时钟信号 128 的上升沿。图 6 更详细地描述了周期积分器模块 500。在这里将乘数数据字 132a 输入到 n 倍模块 600 和加法器 632' 这两个组件之中。这两个组件 600、632 则使用乘数数据字 132a, 以便依据基准时钟信号 128 来分别计算用于第一上升沿即相位 0 以及第二上升沿即相位 2 的所需跃迁。在一个实施例中, n 倍模块 600 将乘数数据字 132a 与 IC 中硬连线的一个常数相乘。如下所述, 对不同条件来说, 所述常数是不同的。所用常数则是出于设计需要而被确定的。

在一个说明性实例中, 选定常数是 2 (用于标准条件)、6 (用于捕获 (catch-up) 模式) 和 11 (用于同步模式)。在标准条件下, 由于相位 0 的计算是对间隔的上升沿进行计算, 因此 n 倍模块 600 会将乘数数据字 132a 与 2 相乘。如下所述, 所述“捕获”是这样一种模式, 其中代数模块 108' 因为某种原因而错误地计算并请求了一个已经出现的需要边沿。这样一来, 在代数模块 108' 能捕获并请求一个将来出现的需要边沿之前, 所述捕获模块 1205 (图 12) 会使得 RISE 时钟 540 以快很多的速度运行。因此需要一个比标准条件中所使用的数字 2 更大的常数, 以便顾及执行捕获的时间。对同步模式来说, 在同步信号之后以及代数模块 108' 之前出现的等待时间周期将会放置一个跃迁边沿。这个等待时间周期表示的是同步模块 304 计算何时放置跃迁边沿来产生“无缝”同步所需要的时间。所述常数必须得到精准的确定, 以便适应于所述等待时间周期。

在标准条件或捕获模式中, n 倍模块 600 的输出发送到加法器 628。所述输出是与用于标准条件或捕获模式之一的校正模式的常数相

乘的乘数数据字 132a。所述校正模式是通过接收自捕获模块 1205 的输入信号 604 确定的,其中所述信号表示的是任意波形生成器 100'是否处于捕获模式。对同步模式来说,作为乘数数据字 132a 与用于同步模式的常数相乘的结果的所述输出将会发送到加法器 624。

在加法器 628 中, n 倍模块 600 的输出添加到寄存器 616 中保存的当前值中。寄存器 616 中保存的当前值表示的是当前相位 0 边沿。在标准条件下,加法器 628 的输出即为当前相位 0 边沿与二倍的依据基准时钟信号 128 的所需周期之和。它表示的是下一个相位 0 的边沿,其中所述边沿是经由复用器 608 传播并在下一个 RISE 时钟信号 540 上计时到寄存器 616 中的。复用器 608 则根据从同步模块 304 接收的转接信号 596 而从加法器 624 或加法器 628 中进行选择。在这里将会重复这个循环(例如将寄存器 616 的内容添加给 n 倍模块 600 的输出并且在下一个 RISE 时钟 540 对其加以保存),以便确定下一个奇数的上升沿跃迁。

相位 2 信号的确定处理(及其对应电路)则稍有不同。寄存器 616 包含了当前相位 0 的边沿。为了计算下一个(例如第二)上升沿,寄存器 616 的输出(也就是第一上升沿)将被发送到加法器 632'并与根据基准时钟 128(也就是乘数数据字 132a)的所需周期相加。加法器 632'的输出表示的是第二(例如偶数)上升沿的值。在下一个反相时钟脉冲 RISEb540'上,加法器 632'的输出计时到寄存器 620'中。反相时钟 RISEb540'则被用于满足用以产生与相位 0 的相差 180 度相位的边沿的相位 2 信道中的严格的定时需要。乘数数据字 132a 能够在每一个(例如第一和/或第二和/或第三等等)上升沿计算中得到改变。因此,所述周期可以随着每一个上升沿而变化,由此产生一个任意波形。

反过来参考图 5,周期积分器模块 500 的输出发送到一个加法器 572。如上所述,周期积分器模块 500 的输出表示的是依据基准时钟 128 而需要下一个奇数跃迁边沿的时间。加法器 572 的另一个输入则是当前时钟时间 508。在下一个奇数跃迁边沿时间中减去从时钟域变换模块 212 接收的当前时钟时间 508,以便确定在下一个奇数跃迁边

沿位置之前所保持的时钟脉冲 128 的数目和部分。如图 6 中所示, 用于下一个偶数跃迁边沿的周期积分器模块 500 的输出发送到一个相同的加法器 572' (未显示)。图 5 所示的用于计算下一个奇数跃迁边沿的电路几乎完全等同于用于计算下一个偶数跃迁边沿的电路。其中的不同之处在于所有寄存器 512'、516'、520'、524'、528'、532'、536' 都是由 RISEb 时钟脉冲 540' 来进行计时的。

用户可以移动所需边沿的相位。这个操作是通过一个第二数据字 132b 来执行的, 其中所述数据字是代数数据输入信号 132 的一部分。在一个实施例中, 第二数据字 132b 是一个 21 比特带符号字, 其中前 8 个比特表示的是基准时钟周期的整数部分, 另外 13 个比特表示基准时钟周期的小数部分。输出信号的相位可以前移 (举例来说, 如果为正的话) 或后移 (举例来说, 如果为负的话) 128 个基准时钟周期之多。如果处于同步模式, 那么所述相移相对的是最后一个同步输入 300。

如果使用第二数据字 132b 来请求一个相移输入, 那么相移将会保存在寄存器 532 中并被添加给加法器 572。加法器 572 的输出则是以基准时钟 128 的周期为单位的超出预发送集合时钟信号 508 的下一个边沿的位置。在每一个 RISE 时钟脉冲 540 上, 加法器 572 的输出时钟输入到寄存器 520 中。由于这个值是用基准时钟 128 衡量的, 因此必须将其转换成与乘法器 576 相乘的环路振荡器 104' 的行程和分支的单位。乘法器 576 则将寄存器 520 的输出 (也就是根据基准时钟 128 的增量边沿位置) 与寄存器 512 的输出相乘, 其中所述输出是当前平均环路速度 (也就是每一基准时钟脉冲 128 的分支)。所述结果则是来自基于环路振荡器 104' 的行程和分支的预发送时钟信号 508 的下一个边沿的时间。此外, 所述结果 (也就是乘法器 576 的输出) 是在 RISE 时钟脉冲 540 上计时输入到寄存器 524 中的。

在一个实施例中, 乘法器 576 是一个 17 比特×17 比特的快速乘法器。并且在一个实施例中, 由于加法器 572 的输出是 21 比特, 因此在加法器 572 与寄存器 520 之间串联使用了一个桶形移位器 (未显示)。

所述桶形移位器从加法器 572 的输出上移到 4 个前端零(例如 MSB), 所移位的零的实际数目依赖于相关的环路速度, 由此将二进制的表示减至 17 个比特。

寄存器 524 的输出表示的是多个行程和分支所预期的边沿时间, 所述输出发送到加法器 580a 并且与环路振荡器 104' 的当前状态相加, 从而确定下一个边沿放置的绝对位置, 在一个实施例中, 所述总和可以是通过添加寄存器 536 中保存的值所移动的另一个相位。保存在寄存器 536 中的值则作为第三数据字 132c 接收, 其中所述第三数据字 132c 是代数数据输入信号 132 的一部分。由于相移是按照分支而不是基准时钟信号 128 进行的, 因此可以使用第三数据字 132c 的输入来精确消除系统中的传播延迟。

加法器 580 的输出发送到加法器 584, 以便与常数 550 相加。常数 550 是通过选择实施方式而被确定的。举例来说, 它可以借助芯片布局而被确定, 并且可以对其进行选择, 以便允许在边沿触发器 1215、1220 上最大限度地建立/保持容限。常数 550 是以已知传播延迟(例如以分支单位)为基础, 由此将选定分支的跃迁集中在其窗口之中。常数 550 只影响到行程的选择而不影响分支的选择。加法器 584 的输出则是在 RISE 时钟脉冲 540 上计时输入到寄存器 528 中的。寄存器 528 中保存的值 560 则是用于选择下一个边沿跃迁的行程和的分支的值 560。寄存器值 560 则发送到切换模块 112、微小延迟模块 310 和输出模块 114, 从而为输出信号 136 中的下一个偶数上升沿(相位 0 信号)跃迁选择正确的边沿跃迁。

同样, 相位 2 信号电路具有一个保存下一个奇数上升沿跃迁的值的寄存器 528'。在一个实施例中, 创建相应偶数与奇数下降沿(相位 1 信号与相位 3 信号)的电路包括一个添加寄存器 528 和 528' 的内容的加法器(未显示)。代数模块 108' 将加法器的输出分为两部分, 以便获取一个与两个上升沿之间中途的时间相对应的值。分离的加法器将会创建偶数和奇数下降沿。在另一个实施例中, 第四数据字 132d(未显示)可以与图 5 描述的相似电路结合使用, 以便根据第四数据字 132d



而将一个下降沿置于偶数与奇数上升沿之间的任何位置。

在计算跃迁边沿位置的过程中,代数模块 108'使用环路平均模块 212 和时钟域变换模块 504。图 7 更详细地描述了环路平均模块 212。环路平均模块 212 接收基准时钟信号 128,以便将时钟信息输入其寄存器 700、712、734、750。环路平均模块 212 还从状态捕捉模块 204'接收所保存的状态信息 220。如上所述,所保存的状态信息 220 包含行程计数器 200 和所述跃迁在对基准时钟 128 的跃迁进行捕捉时传播经过的最后一个分支的状态。环路平均模块对每个基准时钟 128 以分支为单位对环路平均速度进行计算。此外,所述环路平均模块 212 还对最后一个基准时钟脉冲 128 上的环路平均位置进行计算,也就是计算最后一个基准时钟脉冲 128 上的环路瞬时相位。

环路平均模块 212 的平均环路速度部分在减法部件 704 的正输入端接收当前保存的规定信息 220。寄存器 700 则在先前基准时钟脉冲 128 上保存所存储的规定信息 220 的值。减法部件 704 在负输入端接收寄存器 700 中保存的值。减法部件 704 的输出表示所述差值,并且该输出是所述跃迁在一个基准时钟 128 的周期所经过的分支数目。这个值保存在寄存器 712 中。在一个实施例中,算术部件 716、720 和 724 被用于在 64 个环路速度值(例如衰变时间常数=64)上对寄存器 712 中保存的环路速度值求取均值,从而获取一个平均环路速度。在 64 个时钟脉冲上求平均值仅仅是一个设计选择,它是为了提供与合成器的总精度相一致的环路速度精度。在其他实施例中,用于取平均的基准时钟脉冲 128 的数目可以是不同的。

平均速度保存在寄存器 734 中并被提供给时钟域变换模块 504 和同步模块 304。在一个实施例中,如果当前环路速度与平均环路速度在每个基准时钟脉冲 128 上相差八个以上的分支,那么将会使用复用器 730 而在最近的环路速度中切换到寄存器 734 中。这样则允许环路速度发生突然变化,以便复位平均值,并且将可能在启动时出现的很大的瞬变值减至最小。在这里,8 个分支以上的变化是一个可以加以变化、以便满足设计目的的设计选择。

此外,在未曾显示的另一个实施例中还对平均环路速度 564 进行调整,以便补偿为环形振荡器 104 供电的电源的变化。这个变化是由其他电路未曾滤出的频率上的噪声造成的。放大电路(未显示)将这个变化的电压电平缩放成补偿电路可用的电平。补偿电路(未显示)则对电源电压(例如噪声)中经过放大的变化进行测量,并且对此变化进行数字化(例如经由一个模-数转换器来发送噪声信号)。补偿电路则将数字化的变化信号发送到环路平均模块 212,以便调整计算得到的平均环路速度 564。所述变化具有线性相关性(也就是说,电压的增加将会提高环路速度),因此,计算得到的平均环路速度 564 是据此调整的。根据需要,在这里可以使用本领域已知的任何技术来对数字化的变化信号进行缩放。通过在数字化的变化信号中使用更多比特,可以提高补偿的精度。在另一个实施例中,补偿电路将数字化的变化信号发送到代数模块 108,以便补偿电源的变化。

环路平均模块 212 的平均位置(也就是瞬时相位)部分对在分支的小数部分中计算得到的环路位置进行追踪,并且在时间上对所述位置求取均值。计算得到的平均位置则用于增加的精度。环路平均模块 212 的平均环路位置部分在减法部件 738 的正输入端接收当前保存的规定信息 220。减法部件 738 则在负输入端接收计算得到平均环路位置,其中所述位置即为寄存器 750 中保存的内容的输出,先前计算得到的平均环路位置则是在加法器 758 那里与寄存器 734 中保存的平均环路速度相加的。所述差值则即为减法部件 738 的输出,该差值也是实际环路位置与计算得到的平均环路位置之间的差值。在一个实施例中,算术部件 742 和 746 被用于在 32 个值上求取所述差值的平均值,以便为过去的 32 个基准时钟脉冲 128 获取一个平均差值。此外,在这里还可以使用一个不同于 32 的值。而计算得到的平均环路位置则保存在寄存器 750 中。

在一个实施例中,如果当前从实际环路位置中得到的平均环路位置差值大于八个分支,则在先前环路位置中也可以使用复用器 754 来进行切换。由此允许环路相位中出现突然变化,以便复位平均值并将

可能在启动时出现的很大的瞬变值减至最小。八个分支以上的变化是一个设计选择,在这里可以对其加以改变,以便满足设计目的。寄存器 750 的输出或寄存器 700 的输出是根据复用器 754 的状态而在加法器 758 中添加给平均环路速度 564 的,由此创建计算得到的平均环路位置 568。此外,计算得到的平均环路位置 568 还发送到时钟域变换模块 504 和同步模块 304。平均环路速度信号 564 和 `clean_cnt_at_clk` 信号 568 则携带了与其对应的时间常数将要证实的精度比特之外的附加精度比特。通过这个操作,可以避免累积来自系统性取整错误的直流偏压。

图 8 更详细地显示了时钟域变换模块 504。时钟域变换模块 504 从在基准时钟 128 的域中计时的环路平均模块 212 那里接收平均环路速度 564 和计算得到的平均环路位置 568。时钟域变换模块 504 则将基准时钟 128 域中的信号转换成 RISE 时钟 540 域中的信号。时钟域变换模块 504 为在 RISE 时钟 540 域中计时的代数模块 108' 的寄存器 512, 516 (图 5) 提供稳定值。此外,时钟域变换模块 504 还提供当前时钟时间 508, 其中所述时间作为基准时钟 128 的域中的一个计数器 640 开始并且发送到加法器 572, 所述加法器则工作在 RISE 时钟 540 的域中。在所示实施例中,计数器 640 是一个 8 位计数器并且会在达到其最大值时回滚到零。此外,所显示的电路同时还将这些信号校正为 RISEb (例如相位 2 电路)。

`half_clk` 信号 643 是从基准时钟信号 128 中创建的。通过将基准时钟信号 128 分成两个,可以创建 `half_clk` 信号 643。第一层寄存器 644、648、676、680、690、692 是使用基准时钟信号 128 计时的。其中第一层寄存器 644、648、676、680、690、692 用于确保在基准时钟 128 的域中正确捕捉信号 508、564、568。用于各个信号 508、564、568 的电路 (也就是传播路径) 则分别包含了第一层寄存器对 644 和 648、676 和 680、690 和 692。每一对寄存器中的一个寄存器 644、676、690 是在 `half_clk` 信号 643 处于正状态时启用的。这对寄存器中的另一个寄存器 648、680、692 则是在 `half_clk` 信号 643 处于负状态时启

用的。对相位 0 跃迁边沿的计算来说，其中将会使用第二层寄存器中的寄存器 652、682、694。通过使用第一层寄存器对 644 和 648、676 和 680、690 和 692，可以确保这些寄存器对中至少有一个寄存器会在时钟输入到 RISE 时钟 540 的域中的寄存器 652、682、694 的时候是稳定的。即使基准时钟信号 128 是在将 RISE 时钟 540 域的寄存器 652、682、694 用作计时的过程中跃迁的，稳定性也是可以得到确保的。每一个复用器 660、686、698 都会确定使用哪一个第一层寄存器对 644 和 648、676 和 680、690 和 692 作为时钟域变换模块 504 的输出。在这里，所用寄存器是稳定的。复用器的信道选择则是通过寄存器 641 中保存的 half\_clk643 的状态做出的。这个寄存器在 RISE 时钟 540 将数据计时输入 RISE 时钟 540 域的寄存器 652、682、694 之后的某个延迟时间保存 half\_clk 信号 643 的状态。无论何时，只要捕捉所述 half-clk 时所述 half\_clk 正在进行跃迁，那么寄存器 641 和 642' 的输出都是不可预测的。然而，这时所有第一层寄存器都是稳定的，这样一来，只要为 master\_clk 信号 508 和 clean\_cnt\_at\_clk 信号 568 选择相同寄存器，那么所选择的寄存器将是无关紧要的。所显示的延迟则是确保这个属性所必需的。

对相位 2 的计算来说（例如偶数跃迁边沿），在这里使用了相似的电路（也就是寄存器 656'、684'、696'、642' 以及复用器 664'、688'、699'）。其中一个差别在于所有寄存器 656'、684'、696'、642' 都是使用 RISEb 信号 540 来计时的。在所示实施例中，从环路平均模块 212 接收的平均环路速度信号 564 是 21 个比特。桶形移位器 668 和自动排列（range）部件 672 则被用于将所述信号移出多达四个前端比特，以及将所述信号转换成 17 个比特。

如果使用同步信号来确定输出信号 136（也就是同步模式）相位，那么在计算下一个边沿跃迁的时间时将会涉及到同步模块 304。图 9A 更详细地描述了同步模块 304。所述同步模块 304 接收同步输入 300、来自环路平均模块 212 的平均环路速度 564 以及计算得到的平均位置 568。此外，同步模块 304 还接收来自状态捕捉模块 204' 的所保存的状

态信息 330。保存的状态信息 330 包括在对同步输入 300 的跃迁进行捕捉时的行程计数器 200 和最后一个跃迁分支的状态。所述同步模块 304 输出一个表示可以放置无缝边沿跃迁时间的 `start_integrate` 信号 592。由于所述同步模块 304 会在新定相输出能用最小“假信号”来旧定相输出时确定一个准确的半周期，因此所述边沿跃迁将是无缝的。同步模块 304 选择这个跃迁边缘作为指针，以便将输出信号 136 转接成与同步输入 300 同相的信号。同步模块 304 则产生一个转接信号 596，以便向周期同步模块 304 表明同步模块 304 已经计算了所需要的边沿跃迁并且所述边沿跃迁应被代入周期积分器模块 500。

为了计算无缝边沿跃迁位置，同步模块 304 使用减法部件 820 从所保存的状态信息 330 中减去先前基准时钟 128 时计算的平均位置 568。减法部件 820 的输出则表示以分支为单位的基准时钟 128 与同步输入信号 300 之间的相移。在这里使用了三个寄存器 808、812、816 来为两个附加基准时钟 128 的脉冲保持先前基准时钟 128 的脉冲时计算得到的环路振荡器 104' 的平均位置 568，同步去抖动模块 800 则对同步信号 300 的时间进行去抖动操作。

在图 9B 中更详细地显示了同步去抖动模块 800。XOR 部件 904 是一个允许用户设定同步输入信号 300 的极性的可选部件。在这里将 `sync_pol` 的输入设定成同步输入信号 300 的有效变换。在所示实施例中，防抖动电路是为有效上升的同步输入信号 300 而设立的。无论所用同步输入信号 300 的极性如何，所述 XOR 部件 904 都会确保其输出是一个有效上升信号。AND 部件 908 则允许用户启用和禁用同步模式。如果禁用 `sync_enable`（例如低电平状态），那么没有同步脉冲经过所述同步去抖动模块 800。

同步去抖动模块 800 输出一个 `sync_buff` 信号 950。在这里使用了所述 `sync_buff` 信号 950 来对捕捉周期积分器模块 500 的输出的寄存器 832（图 9A）进行计时，从而指示在同步脉冲 300 时出现的相位 0 和相位 2 这两个接下来预期得到的跃迁边沿。Sync-buff 将 `sync` 输入提供到状态捕捉模块 204'。逻辑部件 936 则使得 `sync_buff` 信号

只从同步输入 300 变换的时间持续到触发器 912 的第一 D 输入改变状态之前。此外还添加了延迟 944, 以确保在 D 触发器 912 上, 所述模块 204' 的同步输入是在边沿之前。用于 sync\_buff 信号 950 的时帧则非常窄, 从而允许同步信号在没有出错的情况下任意接近一个 ref-clk128 边沿。一旦寄存器 832 保存所述信息并且第一 D 触发器改变了状态, 那么下一个 D 触发器 916 将会在后继基准时钟 128 的脉冲上改变状态。AND 部件 924 则只切换到高电平状态, 如果同步脉冲(也就是 AND 部件 908 的输出)是高电平, 那么第一 D 触发器 912 将会变成高电平状态, 第二和第三 D 触发器 916、920 则处于低电平状态。如果 AND 部件 924 的输出在后继基准时钟 128 的脉冲上是高电平, 那么第二 D 触发器 915 的输出将会变成高电平状态。由此创建了一个稳定的(也就是去抖动的)同步信号 900 (“SYNCsync”)。在这里不会看到任何持续了不到两个基准时钟 128 的脉冲的同步信号。SYNCsync 信号 900 持续了基准时钟 128 的一个周期。NAND 部件 928、932 与第三 D 触发器 920 则确保所述第二 D 触发器 916 会在 SYNCsync 信号 900 之后的下一个时钟脉冲再次改变状态。当第二 D 触发器 916 再次改变状态时, SYNCsync 信号 900 将会结束。

反过来参考图 9A, 其中将 SYNCsync 信号 900 用于启用寄存器 824。所述寄存器 824 保存减法部件 820 在 SYNCsync 信号 900 之后的下一个基准时钟 128 的脉冲上的输出。此外在这里还使用了所述 SYNCsync 信号 900 来启用寄存器 850, 其中所述寄存器保存的是 SYNCsync 信号 900 之后的下一个基准时钟 128 脉冲上的平均环路速度 564。波纹除法器 804 则将所保存的以分支为单位的相移与寄存器 850 中保存的以每个基准时钟 128 的分支为单位的平均环路速度相除。

上述结果发送到求和器 840 上的一个正输入端。此外还将寄存器 828 中保存的当前时钟时间 508 发送到求和器 840 的一个正输入端。另外还将一个常数 3 发送到求和器 840 的一个负输入端。该常数表示的是用于同步去抖动模块 800 的三个基准时钟 128 的脉冲的等待时间, 并且由此将被减去。此外, 在这里还将当前(也就是转接的时候)相

位 0 边沿位置发送到求和器 840 的一个正输入端。在寄存器 832 中捕捉的相位 0 的边沿位置或是相位 2 的边沿位置都发送到求和器 840 的一个负输入端。复用器 836 则根据与延迟 854 相等的某个时段的 FALLb 信号状态来选择将要使用的信号。这个操作即为在 sync\_buff 信号 950 之前的时钟域之间的异步数据传送。选定的信号表示的是在同步脉冲 300 时计算得到的跃迁边沿。这个信号将会从当前（也就是在转接时）相位 0 的边沿跃迁中减去，以便补偿同步模块 304 的等待时间。而求和器 840 的输出 592 则发送到周期积分器模块 500（图 6）的寄存器 612。所述输出 592 表示的是按照基准时钟 128 的与同步输入 300 同相的输出信号 136 的第一无缝边沿跃迁。

在这里将 D 触发器 850、854、858 用作一个延迟，以确保波纹除法器 804 在转接之前已经固定。而 D 触发器 862、866 则被用于借助转接信号来校准 RISE 时钟 540，其中所述时钟即为周期积分器模块 500 中的寄存器所使用的时钟。D 触发器 866 与 AND 部件 858 确保转接信号仅仅持续 RISE 时钟 540 的一个周期。转接信号 596 则由复用器 608 使用（图 6）。在有效时，复用器 608 对加法器 624 的输出进行选择，其中所述输出包含从同步模块 304 中计算得到的边沿跃迁。在下一个 RISE 时钟 540 的脉冲上，计算得到的边沿跃迁将被计时到寄存器 616 中（图 6），并且转接信号将会变为“无效”。然后，复用器 608 从加法器 628 的输出中进行选择，如上所述，所述输出继续将所需周期长度添加给先前跃迁边沿。

正如上文参考图 5 所描述的那样，用于与同步输出 300 同相的输出信号 136 并且通过按照基准时钟 128 的第一无缝边沿跃迁是从周期积分器模块 500 中输出的。所述跃迁经过加法器 572 和乘法器 576，在所述设备中，所述跃迁从基准时钟 128 的周期转换成分支的数目。然后，边沿跃迁计算经由加法器 580 与加法器 584 继续进行，在所述设备中，所述跃迁在一个 RISE 时钟 540 脉冲上存入寄存器 528。寄存器 528 的输出 560 表示的是用分支和行程来衡量的边沿跃迁位置，并且所述输出将会发送到切换模块 112。切换模块 112 则使用这个信

息来选择环路振荡器 104' 的所需分支, 以便创建下一个边沿跃迁。

图 10A 和 10B 更详细地描述了切换模块 112。图 10A 描述的是用于为相位 0 信号选择边沿的复用器 1000、1003、1010 的结构。在一个实施例中使用了 16:1 的分支选择模块 1003 而从环路振荡器 104' 的 16 个分支中的一个分支选择跃迁边沿。所述分支选择模块 1003 则是由 5 个平衡复用器模块 1000 组成的。

图 10B 更为详细地描述了平衡复用器模块 1000。平衡复用器模块 1000 是一个 4:1 的复用器。所述平衡复用器模块 1000 包括 NAND 部件 1015、1020、1025、1030、1035 以及电容延迟部件 1040、1045、1050、1055。在一个实施例中, 正如本领域已知的那样, 电容延迟部件 1040、1045、1050、1055 是通过使用专门为了在 IC 中创建电容器而设计的 IC 库单元来实现的。电容延迟部件 1040、1045、1050、1055 的电容将会导致当信号在沿相应路径传播时在信号跃迁中出现延迟。平衡复用器模块 1000 接收一个边沿选择输入, 在一个实施例中, 所述输入是一个 2 比特信号, 它定义了的是所选择的四个输入。此外在这里还使用了 NAND 部件 1015、1020、1025、1030、1035 来选择和传播所请求的边沿, 以便进行输出。

对图 10A 中描述的每一个平衡复用器模块 1000a、1000b、1000c、1000d、1000e、1000f、1000g、1000h、1000i、1000j 来说, NAND 部件 1015、1020、1025、1030、1035 都是相同的。然而, 对图 10A 中描述的每一个平衡复用器模块 1000a、1000b、1000c、1000d、1000e、1000f、1000g、1000h、1000i、1000j 来说, 电容延迟部件 1040、1045、1050、1055 则是不同的。这一点是必需的, 因为电容延迟部件 1040、1045、1050、1055 的值基于两个因素, 对每一个平衡复用器模块 1000a、1000b、1000c、1000d、1000e、1000f、1000g、1000h、1000i、1000j 来说, 这两个因素的值并不相同。

第一个因素在于: 正在传播的跃迁究竟是正向 (也就是从低电平状态到高电平状态) 还是负向 (也就是从高电平状态到低电平状态) 跃迁。这两种类型的跃迁都是经由 NAND 部件 1015、1020、1025、



1030、1035 及其路径上的其他部件传播的,但是它们的传播速率则不相同。电容延迟部件 1040、1045、1050、1055 的值可以补偿这两个不同速率。第二个因素在于:边沿从分支到切换模块 112 以及经过切换模块 112 的传播路径具有不同的长度。电容延迟部件 1040、1045、1050、1055 的值可以补偿这些不同的路径长度。无论信号极性或是信号所要传播的路径长度怎样,所述电容延迟部件 1040、1045、1050、1055 的值都会确保跃迁边沿从一个分支传播到切换模块 112 的输出的时间相等。电容延迟部件 1040、1045、1050、1055 的值则依照实施硬件设计而发生变化。此外,这些电容器还校正延迟部件 104 的回路中的不规则性。

在另一个实施例中,平衡复用器模块 1000 的每一个第一电容延迟部件 1040、1045、1050、1055 都具有相应的第二电容延迟部件 1040a、1045a、1050a、1055a (未显示)。每一个相应的第二电容延迟部件 1040a、1045a、1050a、1055a 都与它所关联的第一电容延迟部件并联 (也就是说,1040 与 1040a 是并行的)。同样,第一电容部件集合 1040、1045、1050、1055 会对温度变化、工艺以及电源电压做出反应。而相应的第二电容延迟部件集合 1040a、1045a、1050a、1055a 则不会对温度变化、工艺以及电源电压做出相似的反应。由此确保了可以用相应的第二组电容延迟部件集合 1040a、1045a、1050a、1055a 来补偿相对于温度变化、工艺和电源电压而言并非线性的传播延迟变化。相应的第二电容延迟部件集合 1040a、1045a、1050a、1055a 则是通过一个与实施第一电容延迟部件集合 1040、1045、1050、1055 所不同的 IC 工艺来实现的。在此可以使用本领域已知的任何方法。

回过来参考图 10A,其中使用 and 校准 (也就是选择电容延迟部件的值) 第一分支选择模块 1003a 来从环路振荡器 104' 的分支接收一个上升 (也就是正向) 跃迁。此外还通过使用 and 校准一个第二分支选择模块 1003b (也就是选择电容延迟部件 1040、1045、1050、1055 的值) 来从环路振荡器 104' 的分支中接收一个下降 (也就是负向) 跃迁。如果选定分支是一个下降沿,并且由此所述分支是通过第二分支选择选

择模块 1003b 来选择的,那么所述分支是通过反相器 1005 来反相的。复用器 1010 基于从代数模块 108'接收的输入来选择所用分支选择模块 1003a、1003b。相位 1、相位 2 和相位 3 的信号各自具有相似的复用器结构,以便选择相应的边沿。来自复用器 1010 的上升沿则始终是有有效边沿。

图 11 更详细地描述了微小延迟模块 310。微小延迟模块 310 旨在将所述跃迁边沿延迟一个差分时间量,其中所述时间量小于环路振荡器 104'的分支之间的时间(例如小于图 1B 所描述的 300 皮秒)。微小延迟模块 310 提高了输出信号 136 中的边沿跃迁位置的精度。所述微小延迟模块 310 包括 NAND 部件 1105、1110、1115、1120、1125、1130、1135、1140 以及电容延迟部件 1150、1155、1160、1165、1170、1175。此外,所述微小延迟模块 310 还接收一个边沿选择输入,其中在一个实施例中,所述输入是一个定义了所使用的是哪八条可能路径的三比特信号。NAND 部件 1105、1110、1115、1120、1125、1130、1135、1140 则被用于选择所请求的边沿并且将其传播到输出端,由此可以经由八个可能路径之一来对其进行路由。任意波形生成器 100'包括一个用于代数模块 108'所计算的各条信道的微小延迟模块(例如相位 0、相位 1、相位 2 和相位 3)。所关注的边沿则总是正向进入和离开所述微小延迟模块 105。

如上所述,对每个微小延迟模块 310 来说,电容延迟部件 1150、1155、1160、1165、1170、1175 可以是不同的。然而在优选实施例中,微小延迟模块的内部设计全都是相同的,由此将会导致具有相同的电容值。此外如上所述,每一个微小延迟模块都可以包括相应的电容延迟部件集合 1150a、1155a、1160a、1165a、1170a、1175a。在一个实施例中,通过微小延迟模块 108 的快速路径甚至远远大于每一个分支之间的延迟时间。然而在这八条可能路径中,任何两条路径之间的延迟时间差值都小于各个分支之间的延迟时间。微小延迟模块添加的很大的固定延迟分量则是由代数模块 108'所使用的常数 550(图 5)的一部分以及代数模块 108'中的相位补偿输入 132c 的一部分(可选的)来

进行补偿的。而所述微小延迟模块 310 的输出则发送到输出模块 114。

在图 12 中更详细地描述了输出模块 114。所述输出模块包括一对边沿触发器 1215、1220 以及一个窗口模块 1210。这三个组件 1215、1220、1210 是对应于各条信道（例如相位 0、相位 1、相位 2 和相位 3）而被包含的。此外，输出模块 114 还包括相位或信道组合模块 1225。用于各条信道的边沿触发器对 1215、1220（例如相位 0、相位 1、相位 2 和相位 3）的输出则发送到相位或信道组合模块 1225。相位或信道组合模块 1225 将所有信道信号（例如相位 0、相位 1、相位 2、相位 3）组合成一个单独输出 136。如果需要用到多个输出，那么可以通过相应信道组合模块 1225 来为各个输出合并不同信道组合。输出模块 114 还包括一个捕获模块，在一个实施例中，所述捕获模块只包括上升沿信道（例如相位 0、相位 2）。

在图 13 中更详细地描述了捕获模块 1205、窗口模块 1210 以及一对边沿触发器 1215、1220。在这里将接收自微小延迟模块 310 的相位 0 边沿用作这两个边沿 JK 触发器 1215、1220 的时钟。这对边沿触发器 1215、1220 的 JK 输入则是从窗口模块 1210 中接收的。在一个实施例中，窗口模块 1210 具有两个比较器 1336、1360。第一比较器 1336 将代数模块 108' 计算的行程与同相行程计数器 200a 相比较。第二比较器 1360 则将代数模块 108' 计算的行程与异相行程计数器 200b 相比较。当代数模块 108' 计算的行程与行程计数器 200 相等时，所述窗口打开。在这里将边沿触发器对设定成通过 AND 部件 1376 或 1382 来对源自微小延迟模块 310 的边沿跃迁输入进行计时。窗口开启触发器 1340 或触发器 1372 中的每一个触发器都是基于以代数方式计算的单个比特并由复用器 1344 使用和选择的。此外还使用 XOR 部件 1348、触发器 1352 以及 NAND 部件 1356 在所需边沿跃迁成功计时了边沿触发器对 1215、1220 的时候复位并关闭窗口。当得到代数信号 `ph0_win_open` 的创建替换窗口的指示时，触发器 1368 和 AND 部件 1364 将被用于创建一个替换窗口。替换窗口则用于使得波形合成器能以很高频率工作，例如在高于自由振荡环路本身的频率上工作。在这种情况下，所

需环路已经处在进行中，使用包括比较器 1350、1336 的机制则变得过晚。取而代之的是，除了延迟之外，替换窗口将会立即“打开”，以使相关数据有效。

第一比较器 1336 还确定代数模块 108' 计算的行程是否比行程计数器 200a 的当前值与等待时间的容限之和更小。如果小于的话，那么代数模块 108' 传送一个错误计算并且第一比较器会将一个信号发送到捕获模块 1205。所述捕获模块 1205 在代数模块 108 请求一个过去发生的行程（也就是所请求的行程小于当前行程）的时候有效。比较器 1336 的输出表示所请求的是一个过去的行程，该输出发送到捕获模块 1205 的触发器 1308。在环路振荡器 104' 的下一个周期，误差信号将被计时输入到触发器 1308 中。采用 ph0\_catch 标记的触发器 1308 的输出则发送到一个 OR 部件 1312。相位 2 信号具有一个相似的捕获模块 1205'（未显示）。其中的一个不同之处在于，触发器部件 1309 和 OR 部件 1310 没有与相位 2 捕获模块 1205' 包含在一起。而相位 1 和相位 3 信号则不具有捕获模块 1205。对相位 2 的捕获模块 1205' 来说，其触发器 1308'（未显示）的输出也是 OR 部件 1312 的输入以及触发器 1309 的输入。在触发器 1308' 的输出发生变化之后的 RISEb 时钟 540' 的脉冲上，触发器 1309 向 OR 部件 1310 输出一个有效捕获信号。所述 OR 部件 1310 则将一个有效捕获信号 604 输出到周期积分器模块 500 的 n 倍模块 600。如上所述，所述 n 倍模块 600 使用这个信号 604 来确定所述任意波形生成器是否处于捕获模式之中并且是否应用了正确的乘法器。在处于捕获模式时，通过使用边沿触发器对 1215、1220 的置位和复位输入，所述 NAND 部件 1316、1320、1324、1328 以及 AND 部件 1332 将被用于以四分之一的自由环路振荡器 104' 的速度来创建 RISE 时钟 540 的脉冲。如上所述，在将来再一次请求所述行程之前，所述捕获模式将会为代数模块 108' 创建一个替换高速时钟并增加经过编程的周期。而在那时，所述捕获模块 1205 将会结束捕获模式（例如将捕获信号 604 改成“非”有效状态）。在电路恢复时，所述电路将以正确的相位恢复的。

参考图 14, 输出模块 114 的相位或信道组合模块 1225 包含三个平衡 XOR 模块 1500a、1500b、1500c 以及四个反相驱动部件 1405、1410、1415、1420。如上所述, 前两个平衡 XOR 模块 1500a、1500b 的输出从输出模块 114 发送到代数模块 108' 并且由代数模块 108' 用作执行计算的时钟。此外, 这些输出还发送到第三平衡 XOR 模块 1500c, 以便创建输出信号 136。第一平衡 XOR 模块 1500a 接收上升沿信道信号 (也就是相位 0, 相位 2)。接下来的平衡 XOR 模块 1500b 则接收下降沿信道信号 (也就是相位 1, 相位 3)。在这里使用了相位组合模块 1225 来合并每一个信号 (也就是相位 0, 相位 1, 相位 2 以及相位 3)。所述相位组合模块 1225 是以一种平衡方式进行组合的, 这样一来, 无论信号或信号极性怎样 (也就是相位 0, 相位 1, 相位 2, 相位 3), 来自边沿触发器对 1215、1220 的输出端到输出信号 136 的传播延迟时间几乎都是相同的。

图 15 更详细地描述了平衡 XOR 模块 1500。平衡 XOR 模块 1500 包括 6 个 NAND 部件 1505、1510、1515、1520、1525、1530 以及八个反相器 1535、1540、1545、1550、1555、1560、1565、1570。平衡 XOR 模块 1500 还包括电容延迟部件 1575a、1575b、1575c、1575d、1575e、1575f、1575g、1575h。对图 14 中描述的每一个平衡 XOR 模块 1500a、1500b、1500c 来说, 电容延迟部件 1575a、1575b、1575c、1575d、1575e、1575f、1575g、1575h 都是不同的。如上所述, 对补偿用于各个输入的不同路径长度以及这些输入处理是上升沿还是下降沿跃迁来说, 电容延迟部件 1575a、1575b、1575c、1575d、1575e、1575f、1575g、1575h 并不相同。此外如上所述, 每一个平衡 XOR 模块都可以包括一个相应的第二电容延迟部件集合 1575aa、1575ab、1575ac、1575ad、1575ae、1575af、1575ag、1575ah。

作为一个说明性实例, 图 16 描述的是输出一个输出信号 136" 的任意波形生成器 100" 的时序图, 其中所述信号是一个周期为 7 毫微秒 (也就是大约 143MHz) 的方波。这个时序图描述的是一个任意波形生成器 100", 其中代数模块 108" 包含了用于计算两个交替上升沿 (也

就是相位 0 和相位 2) 以及两个交替下降沿 (也就是相位 1、相位 3) 的并行电路。此外, 所述时序图还显示了从输入端进入切换模块 112" 并且到达输出模块 114" 的输出端的各个信号。

标记了“16:1 复用器”的方框表示的是经过分支选择模块 (例如如图 10A 的分支选择模块 1003a) 的传播时间。标记了“符号”的方框表示的是经过所述在正 (也就是上升沿) 分支选择模块 (例如如图 10A 中的 1003a) 或负 (也就是下降沿) 分支选择模块 (例如如图 10A 中的 1003b) 之间进行选择的复用器符号选择部件例如如图 10A 中的复用器 1010 的传播时间。标记了“微调”的方框表示的是经过微小延迟模块 (例如如图 11 的微小延迟模块 310) 的传播时间。微调方框之前的边沿上的双线表示的则是经由微小延迟模块 310 传播的变量。标记了“JK flops”的方框表示的是经过边沿触发器对 (例如如图 13 中的触发器 1215, 1220) 的传播时间。紧随在标记了“JK flops”的方框之后并且标记了“XOR 的方框表示的是经过第一平衡 XOR 模块 (例如如图 14 所示的用于相位 0 和相位 2 信号的平衡 XOR 模块 1500a 或者用于相位 1 和相位 3 信号的平衡 XOR 模块 1500b) 的传播时间。紧随在标记了“XOR”的方框之后并且标记了“XOR”的下一个方框表示的是经过下一个平衡 XOR 模块 (例如如图 14 的平衡 XOR 模块 1500c) 的传播时间。标记了“时钟树”的方框表示的是经过时钟树 (例如如图 14 中所示的时钟树) 的传播时间。标记了“out\_clk”的信号表示的是由“任意波形生成器 100 输出的输出信号 136”。

本发明可以在不脱离本发明实质或基本特征的情况下采用其他特定形式来加以实施。因此, 在各个方面都将上述实施例视为是说明性的, 而不是对这里描述的发明加以限制, 这样一来, 本发明的范围是通过附加权利要求而不是上文的描述来进行限定的, 并且这其中意图包含所有那些落入权利要求等价物的含义和范围以内的变化。

图1A

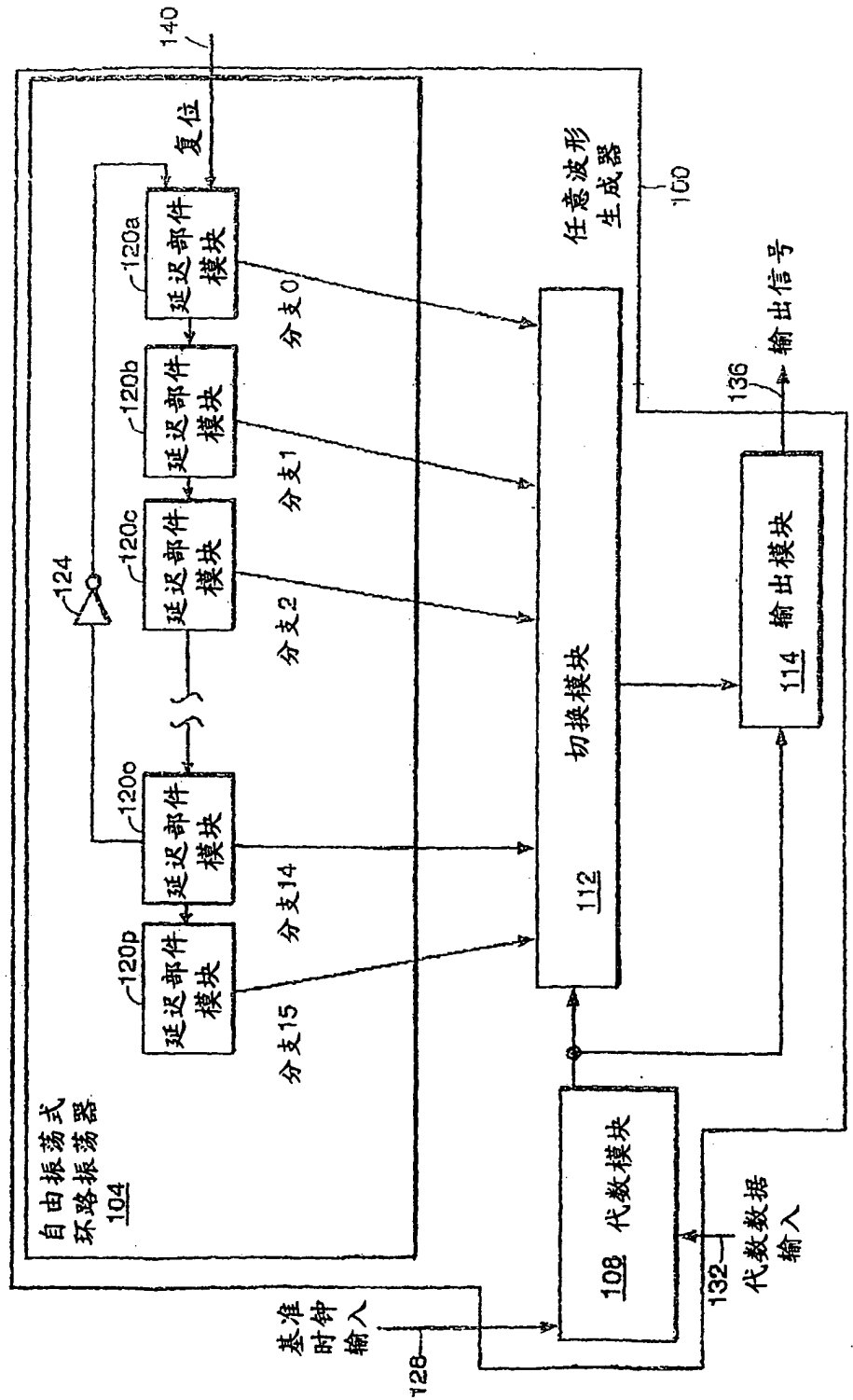


图1B

时序图-自由振荡式振荡器的延迟输出

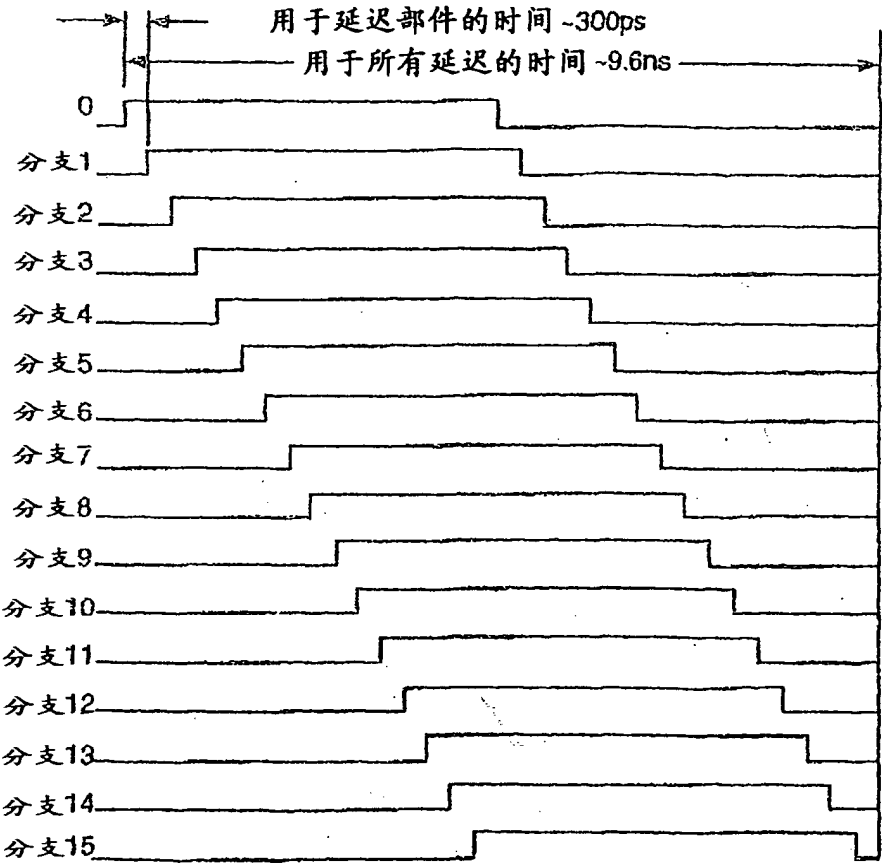




图 2

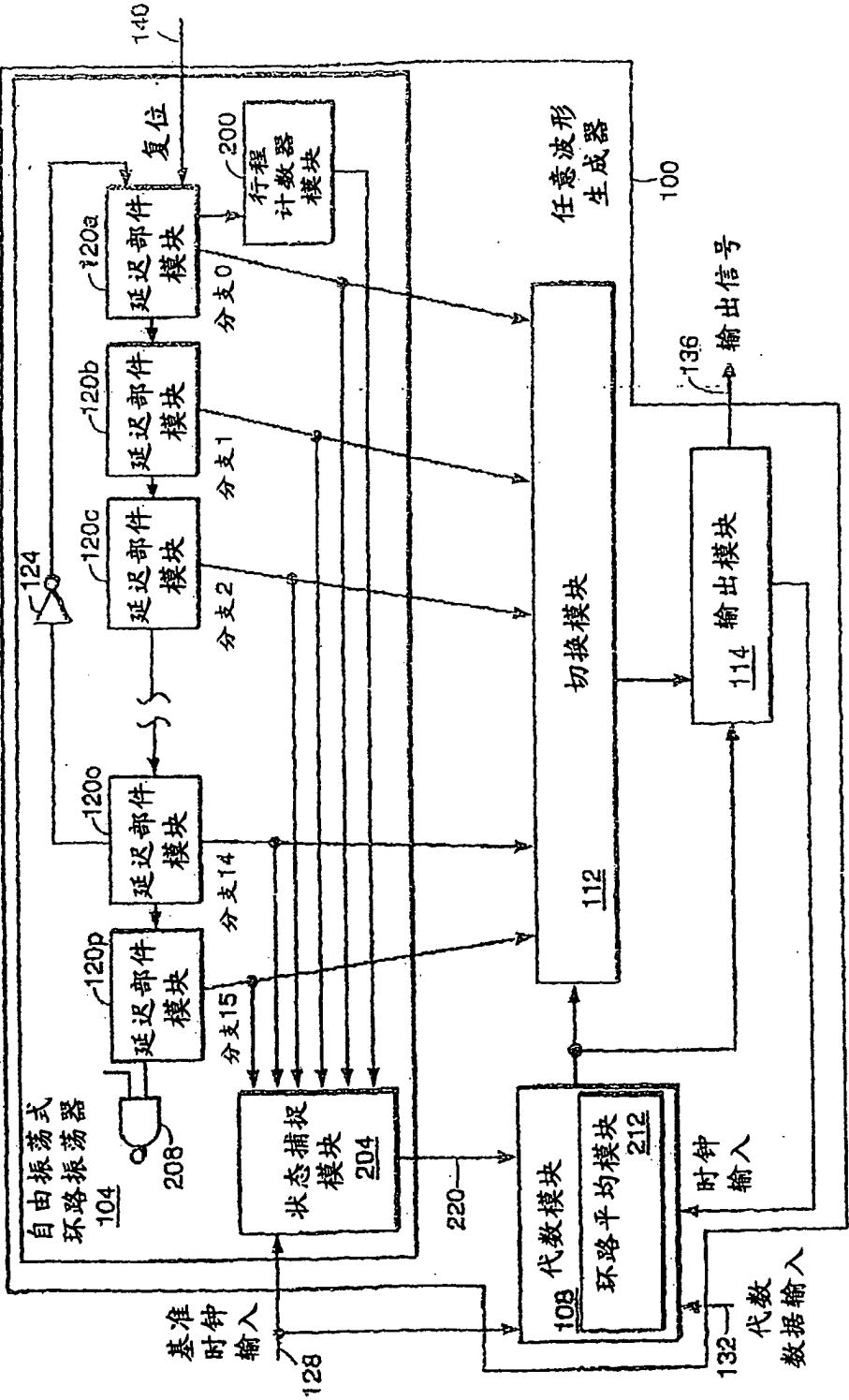


图3

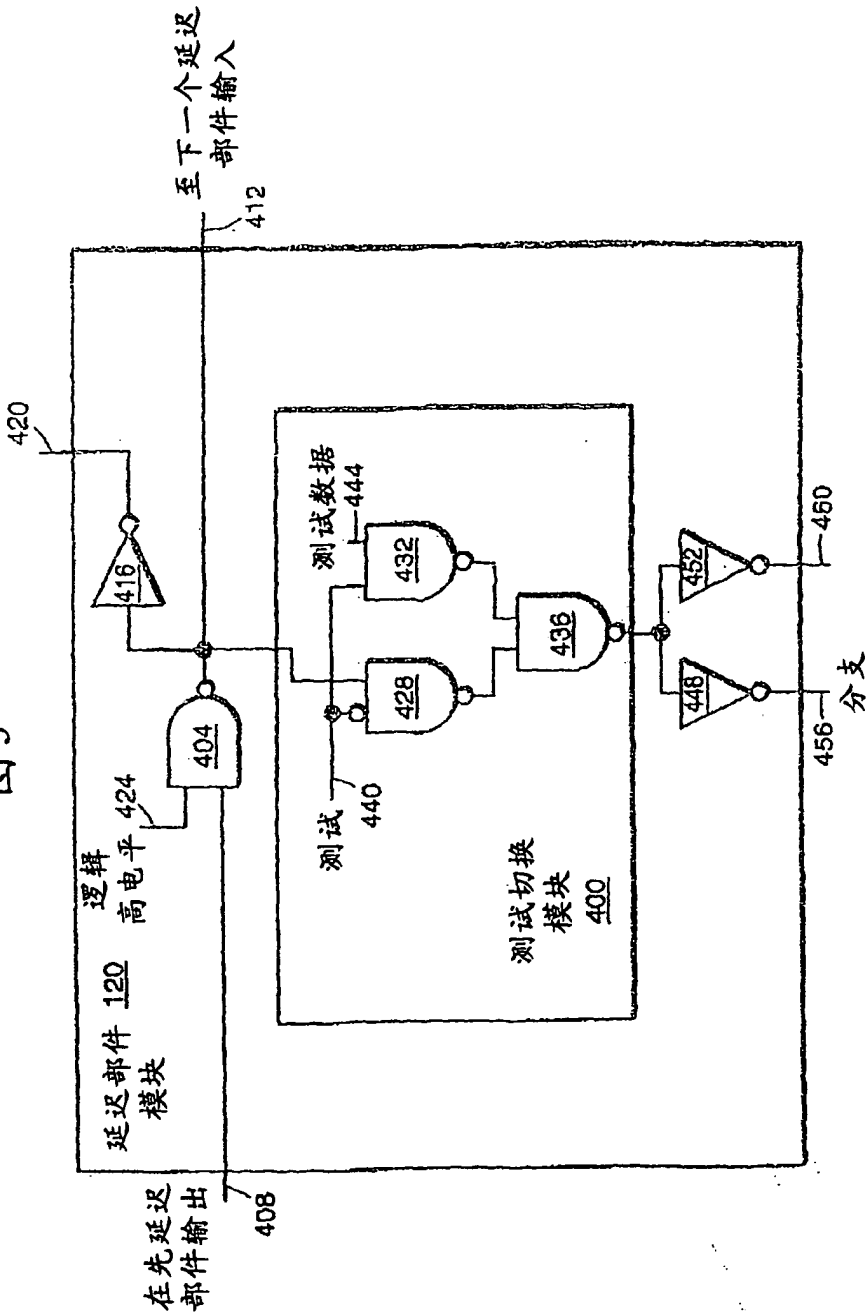


图4

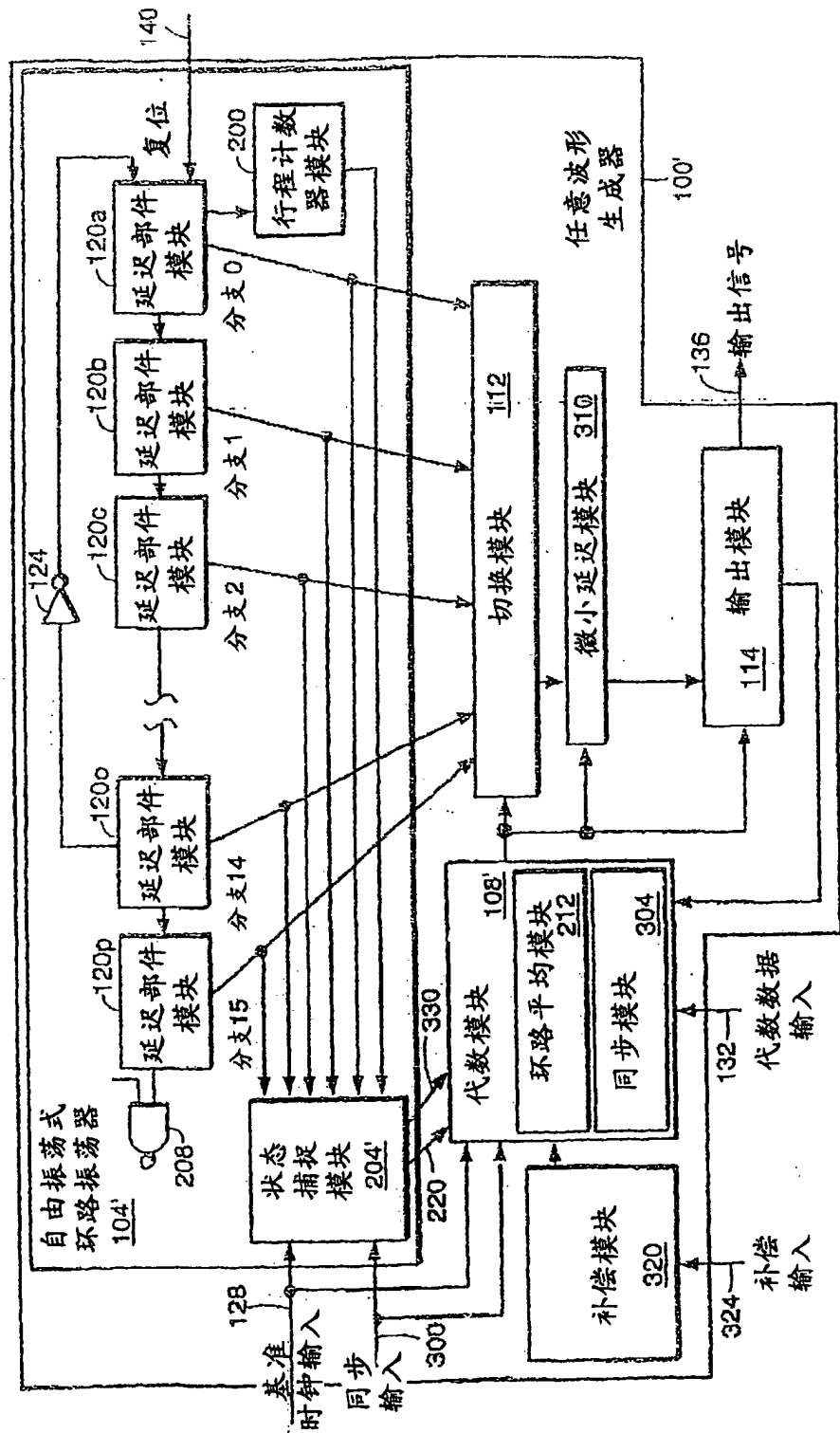
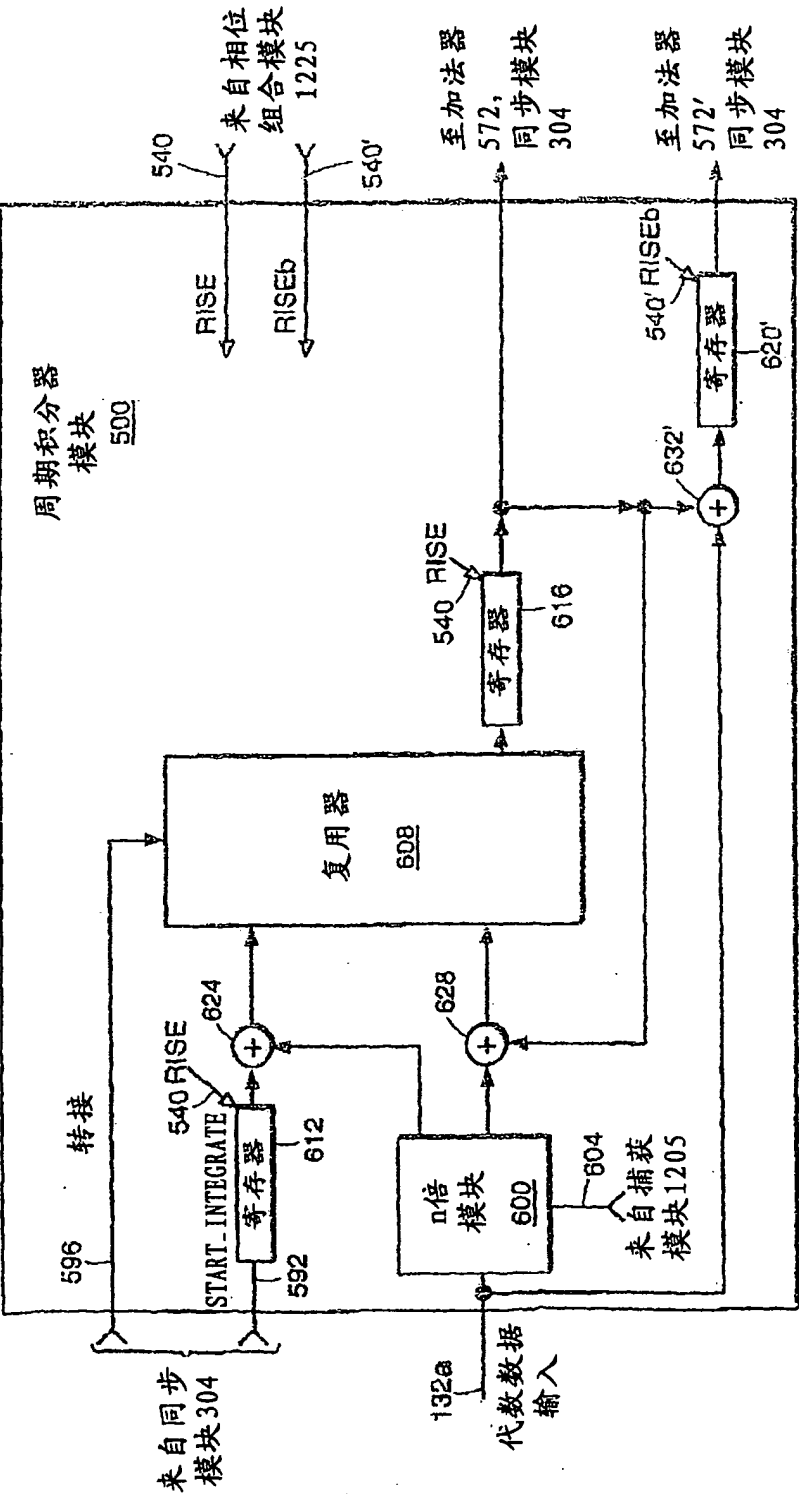


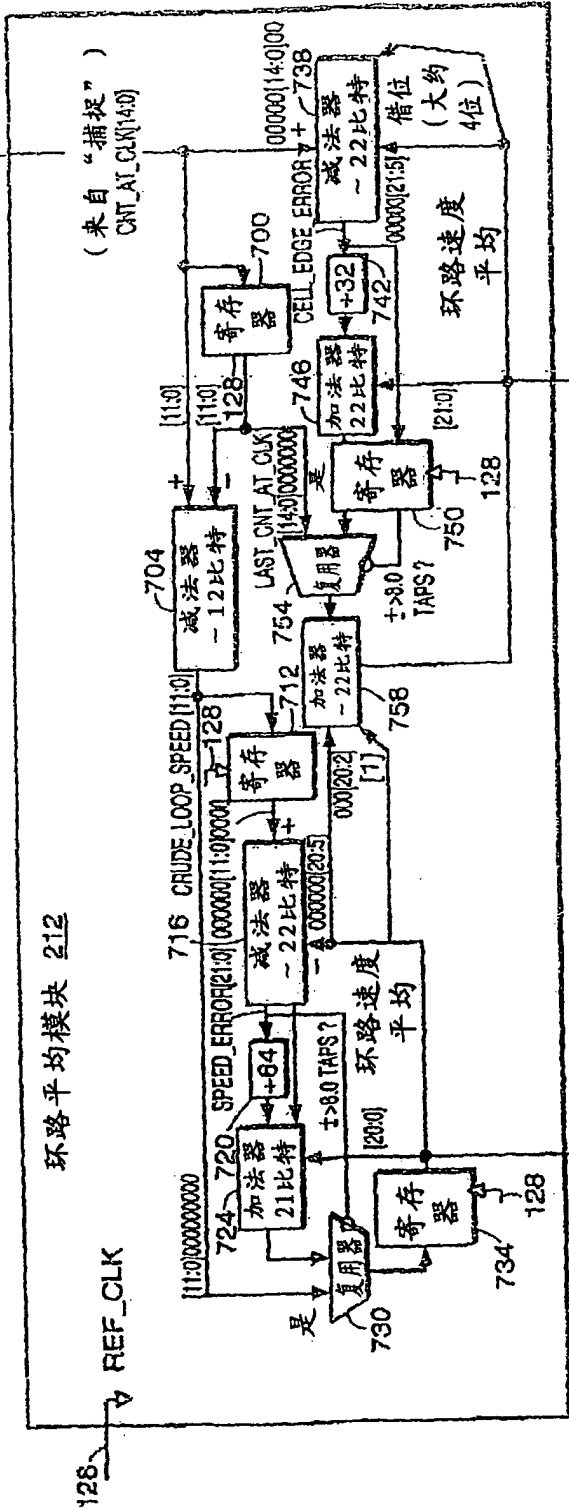


图6



来自状态捕捉  
模块204

图 7



CLEAN\_CNT\_AT\_CLK [21:0]  
单位: (15.7) 分支/时钟  
至时钟域变换模块504,  
同步模块304

AVG\_LOOP\_SPEED [20:0]  
单位: (12.9) 分支/时钟  
至时钟域变换模块504,  
同步模块304



图 9A-1

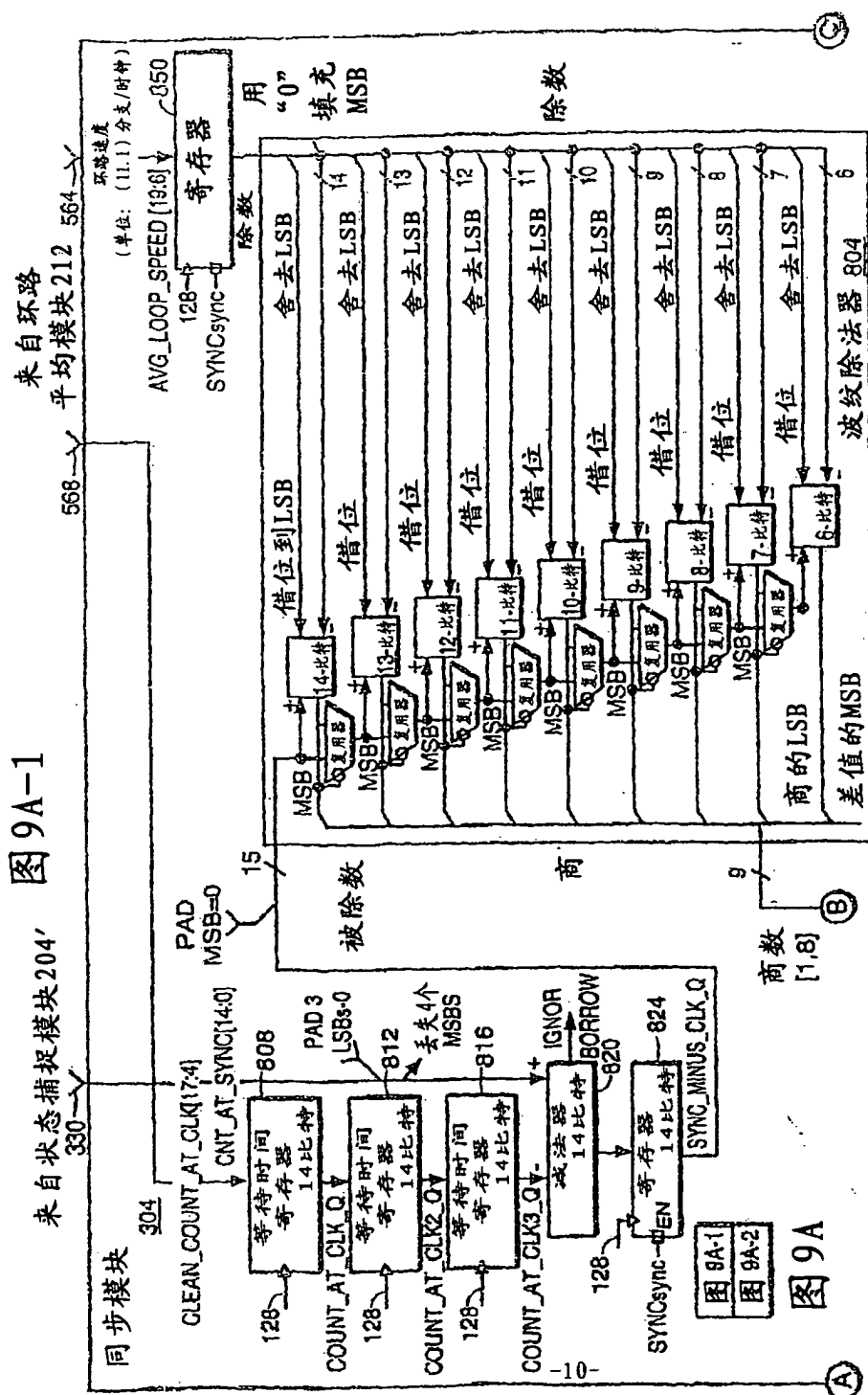




图 9A-2

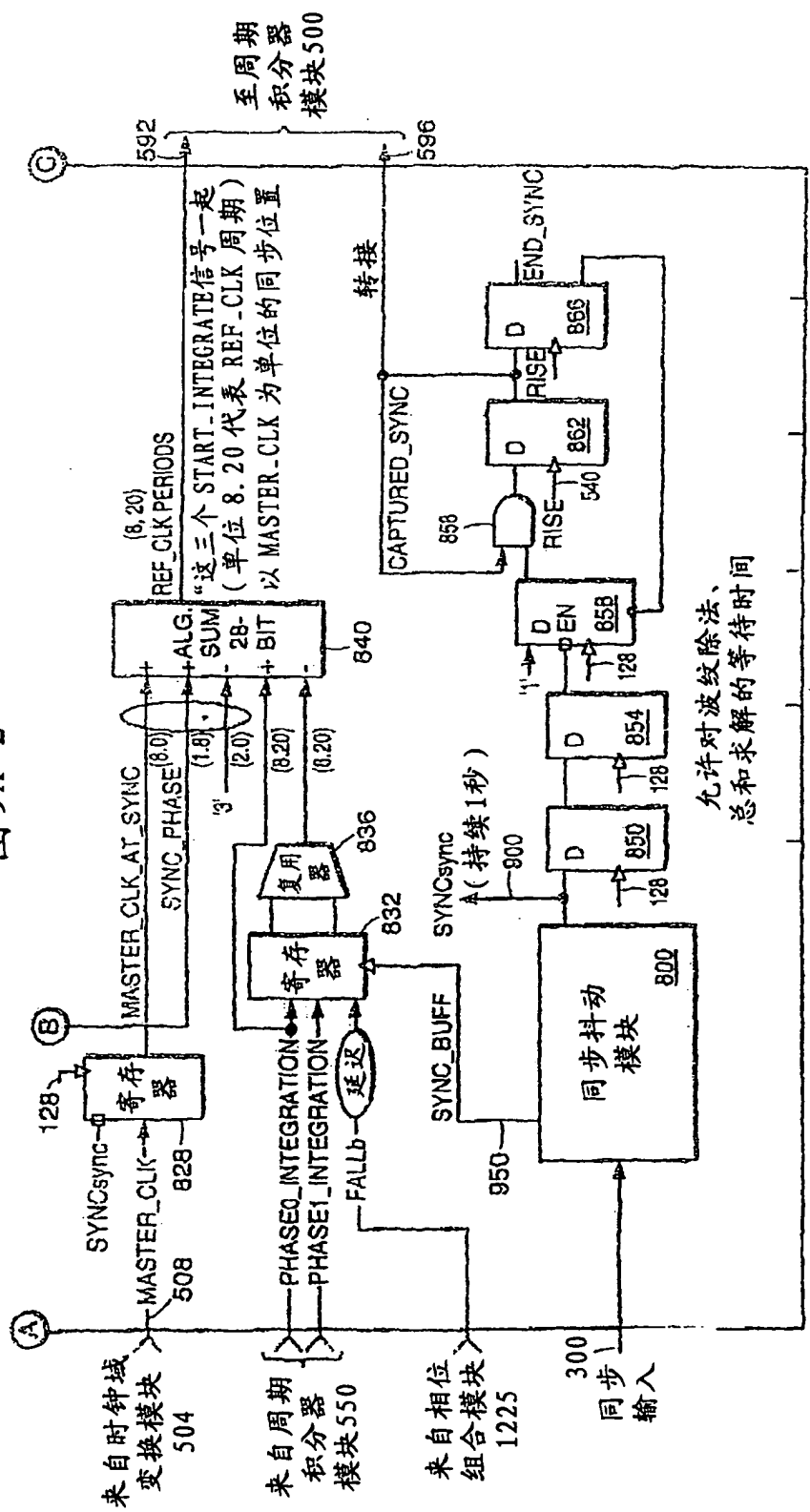


图 9B

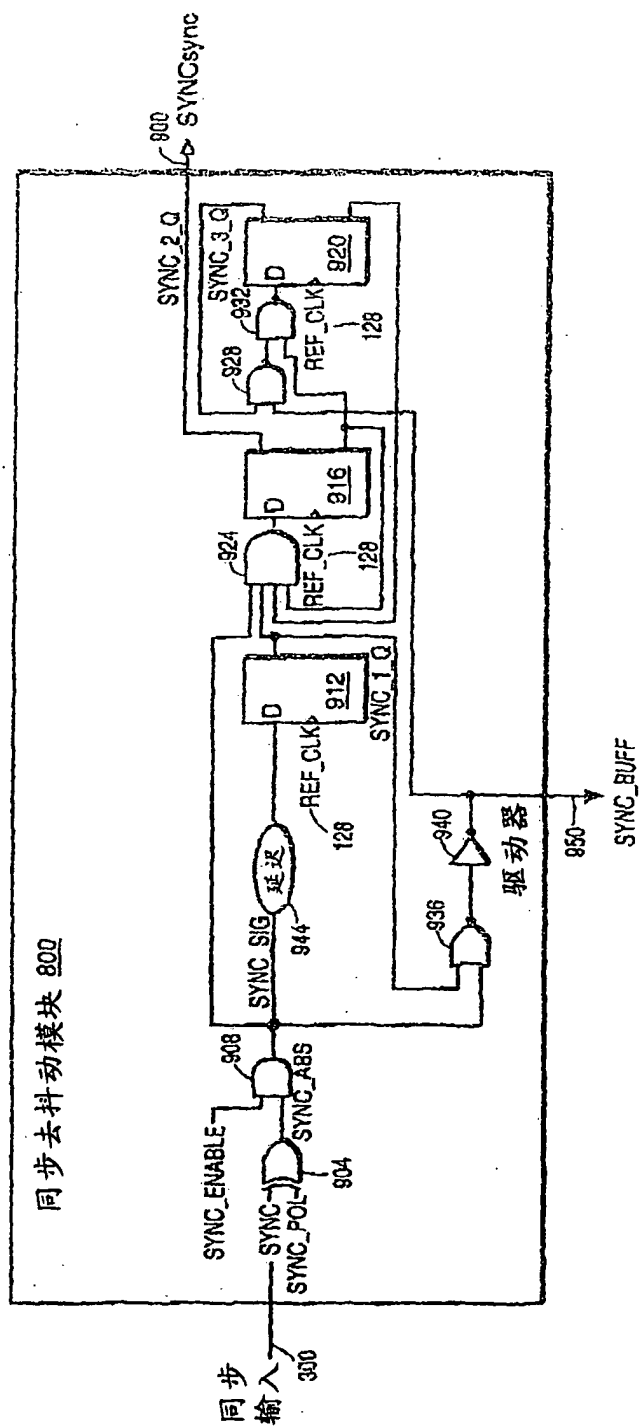


图 10A

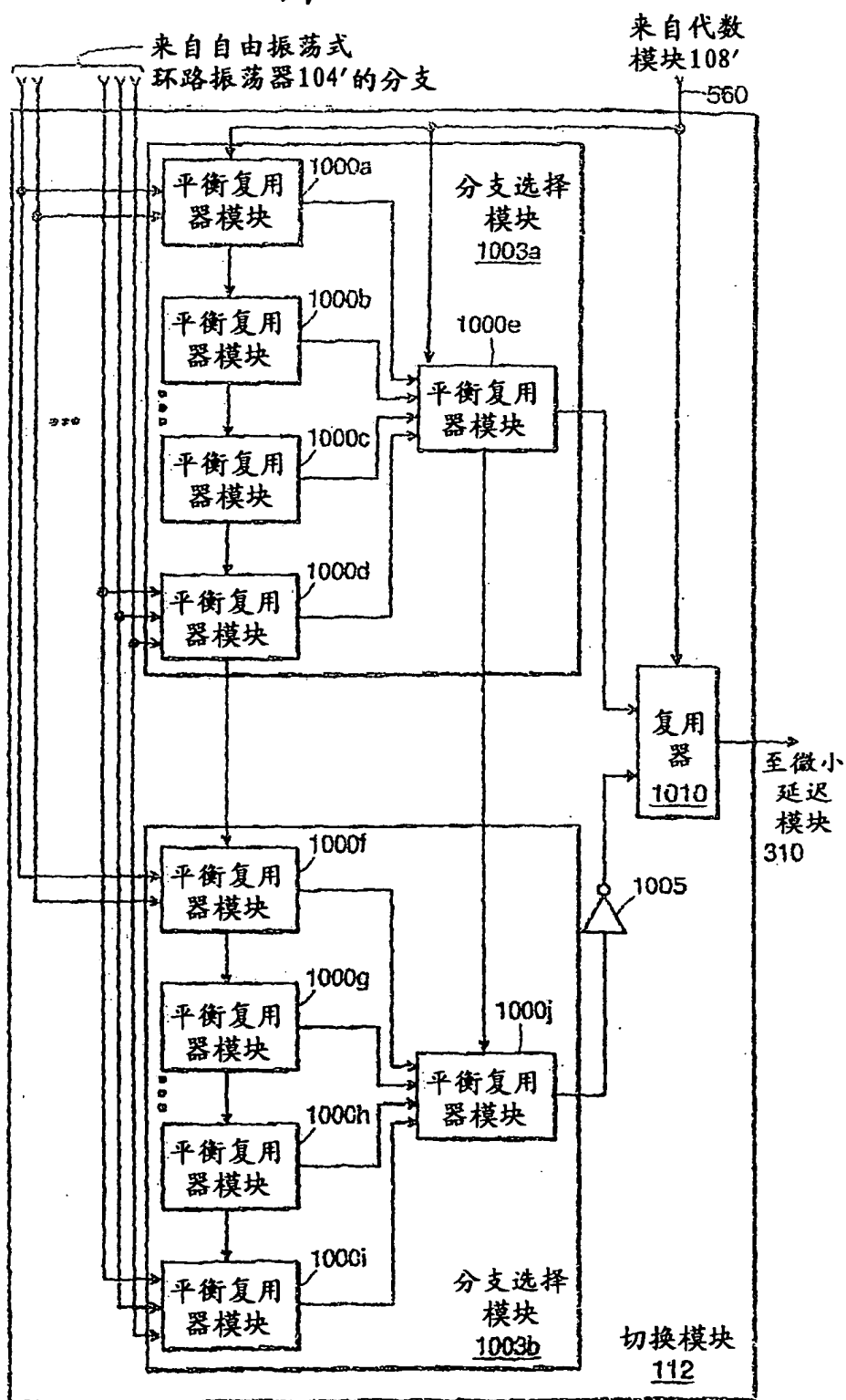


图10B

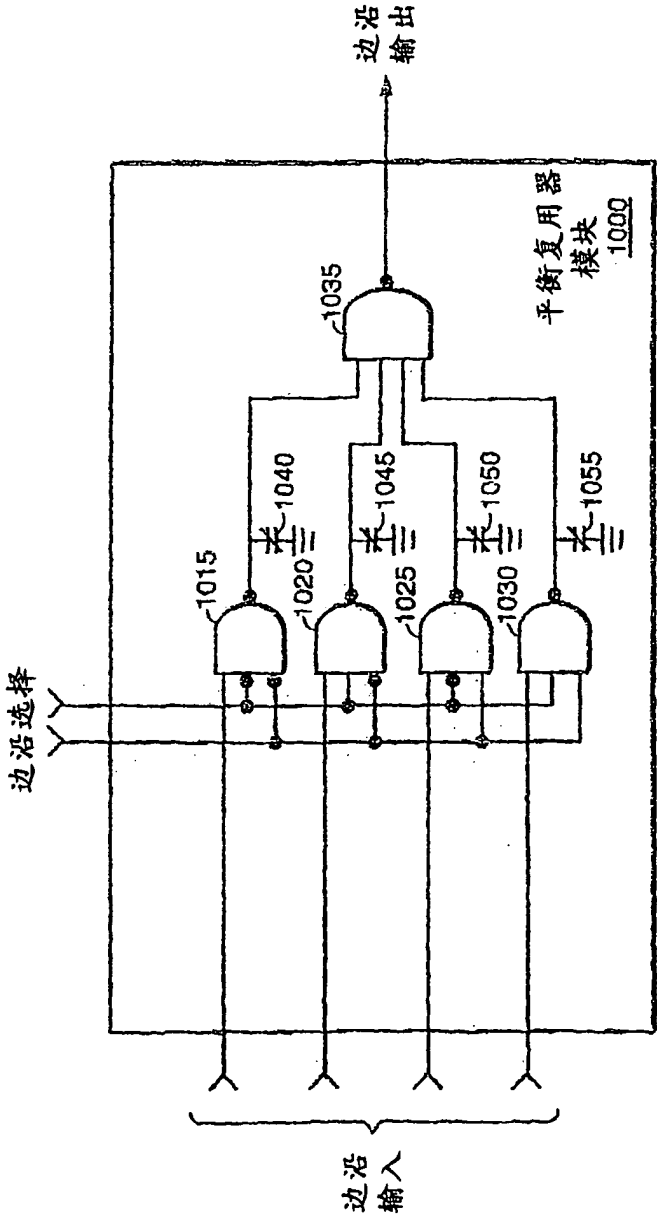


图 11  
来自代数模块108

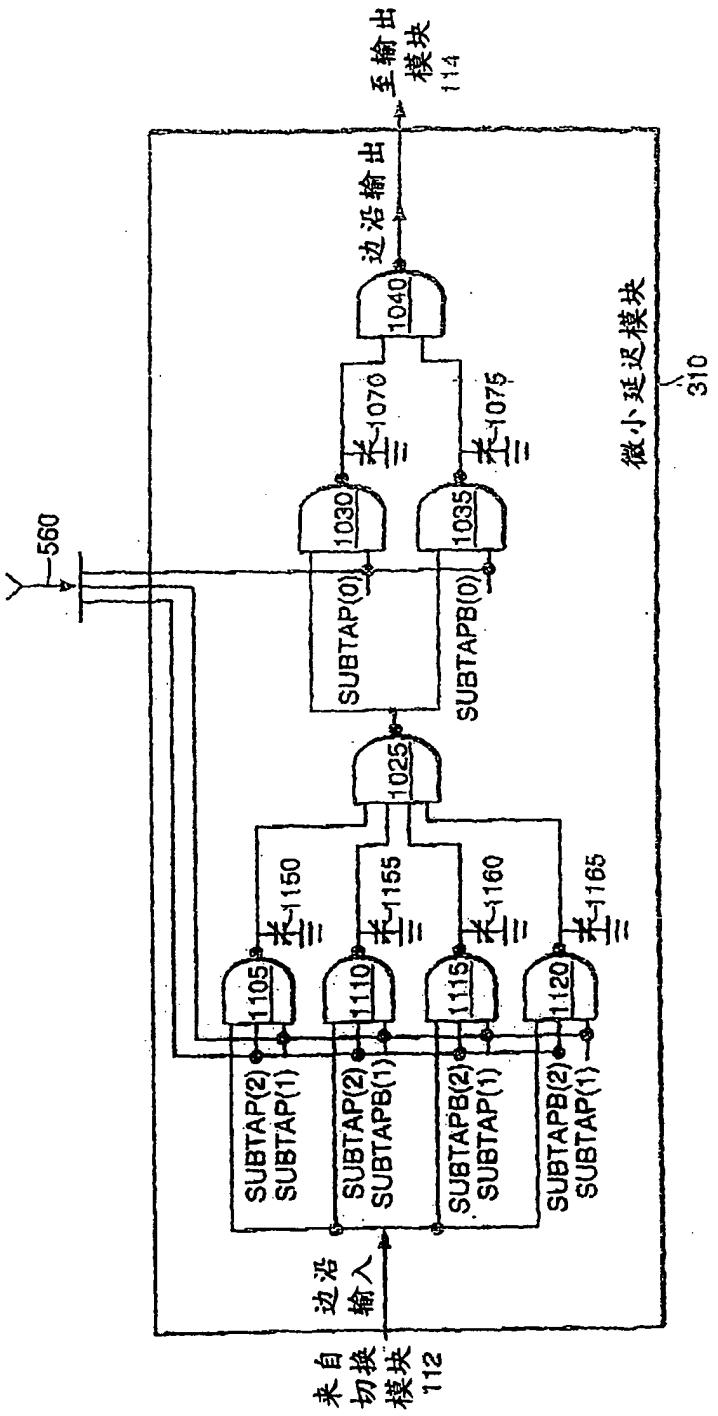


图 12

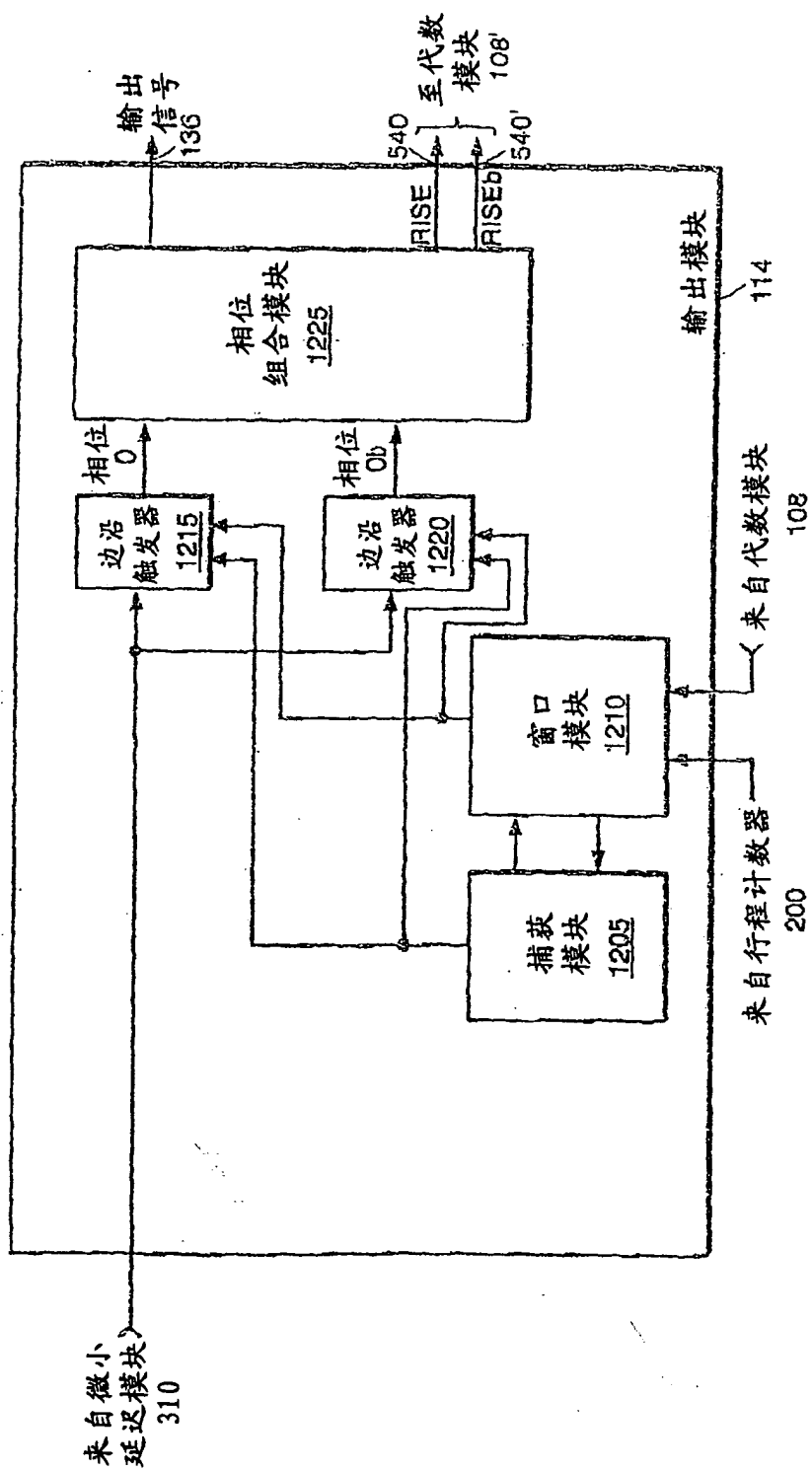


图13

图13A

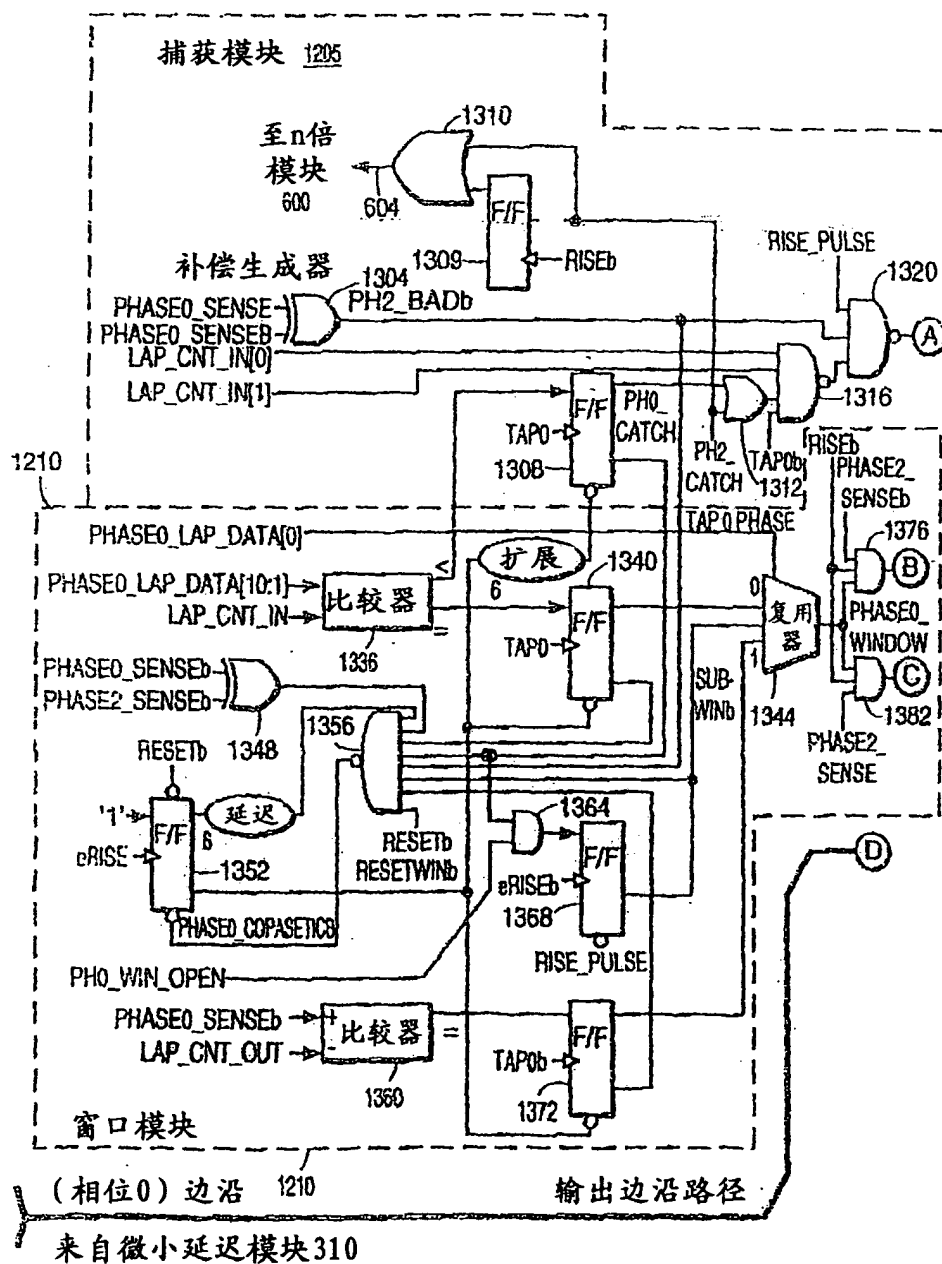
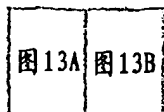


图 13B

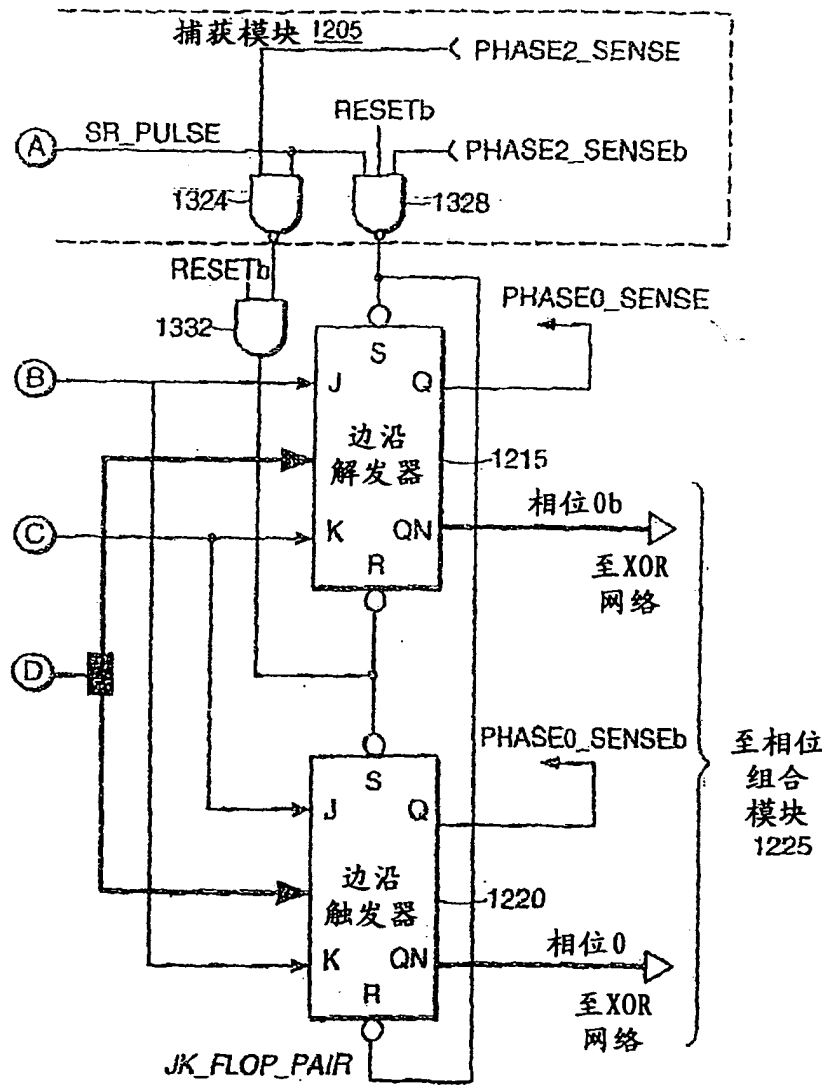




图 14

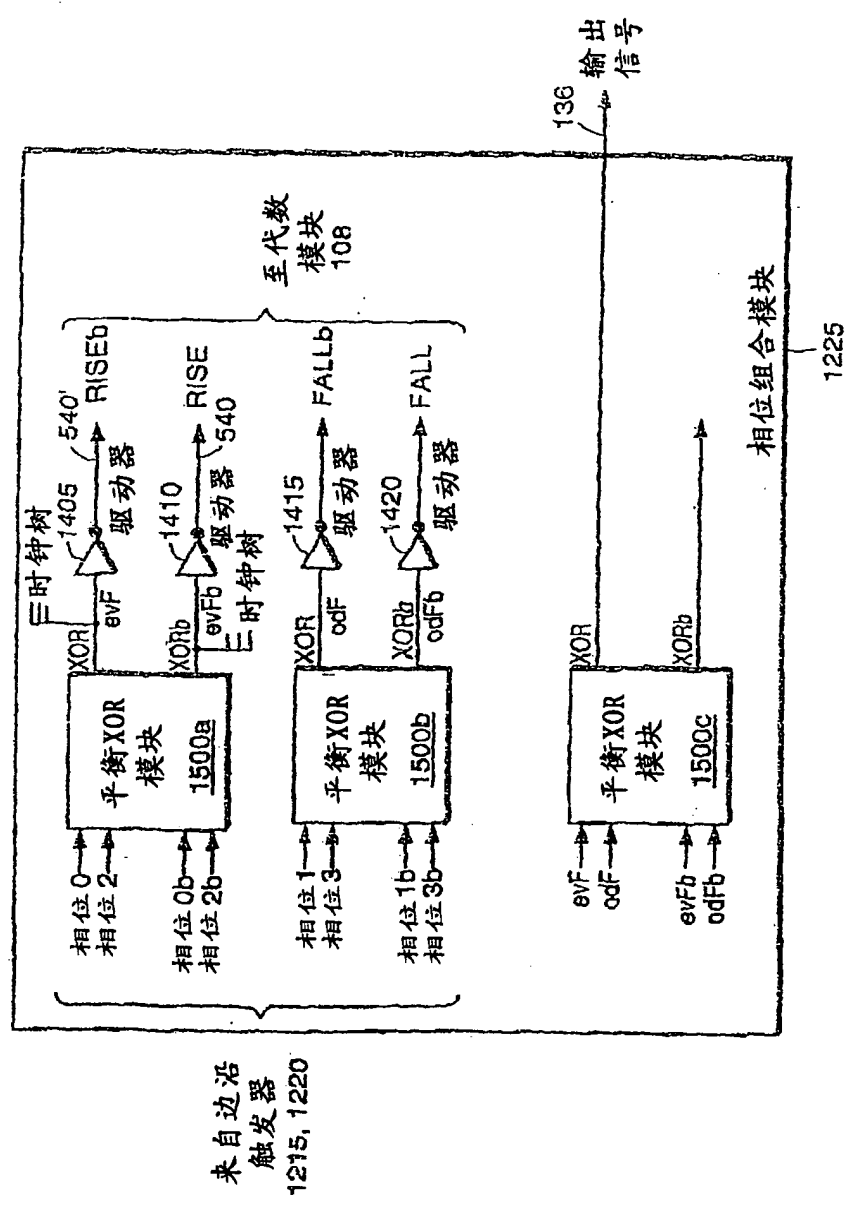


图 15

平衡XOR模块

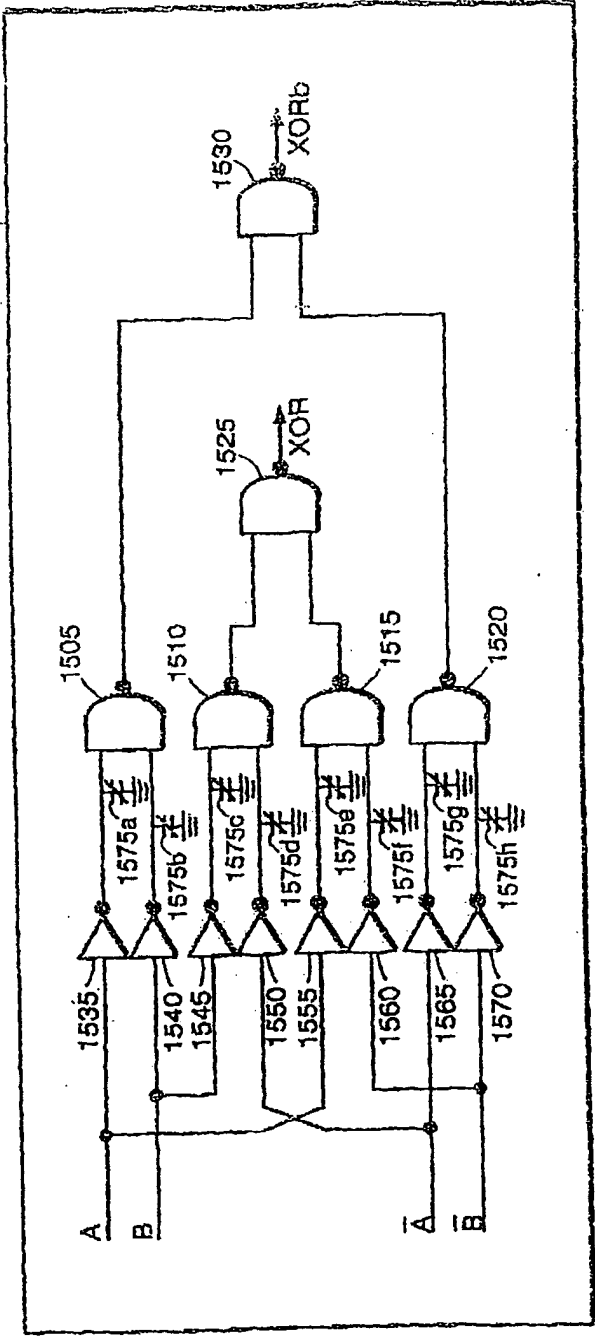


图 16A

所描述的输出周期: 7ns [143MHz]  
所显示的最坏情况下的传播

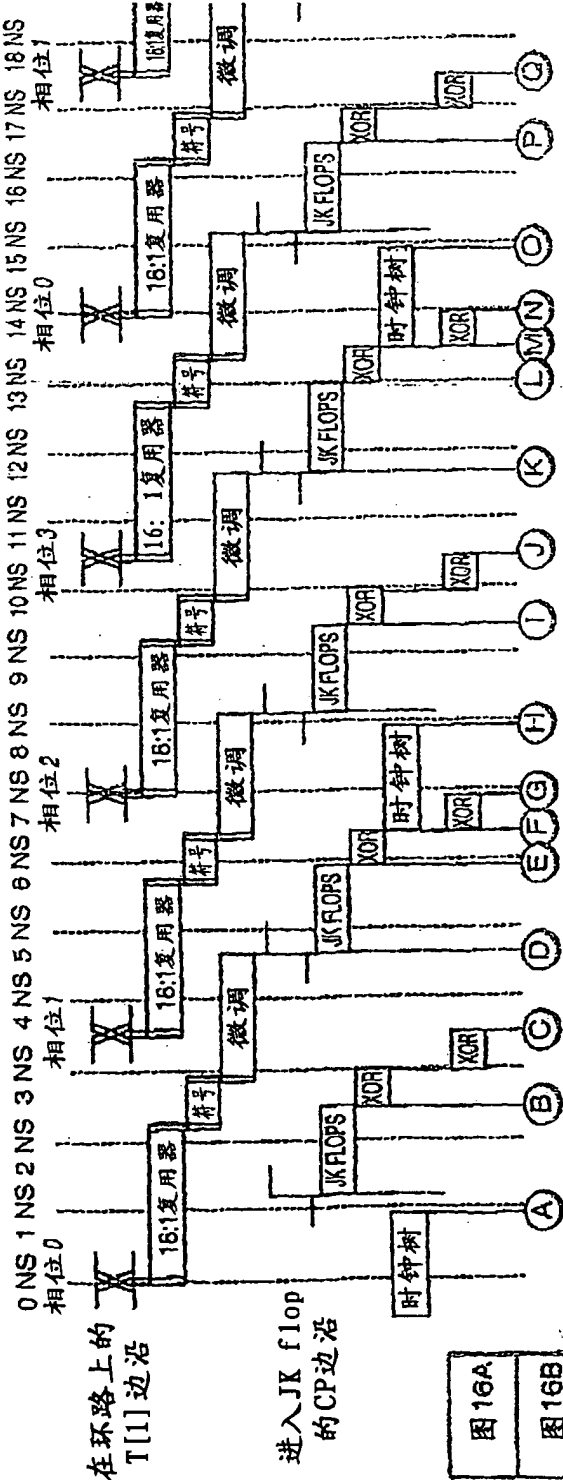


图 16A
图 16B
图 16C
图 16D

图 16

图 16B

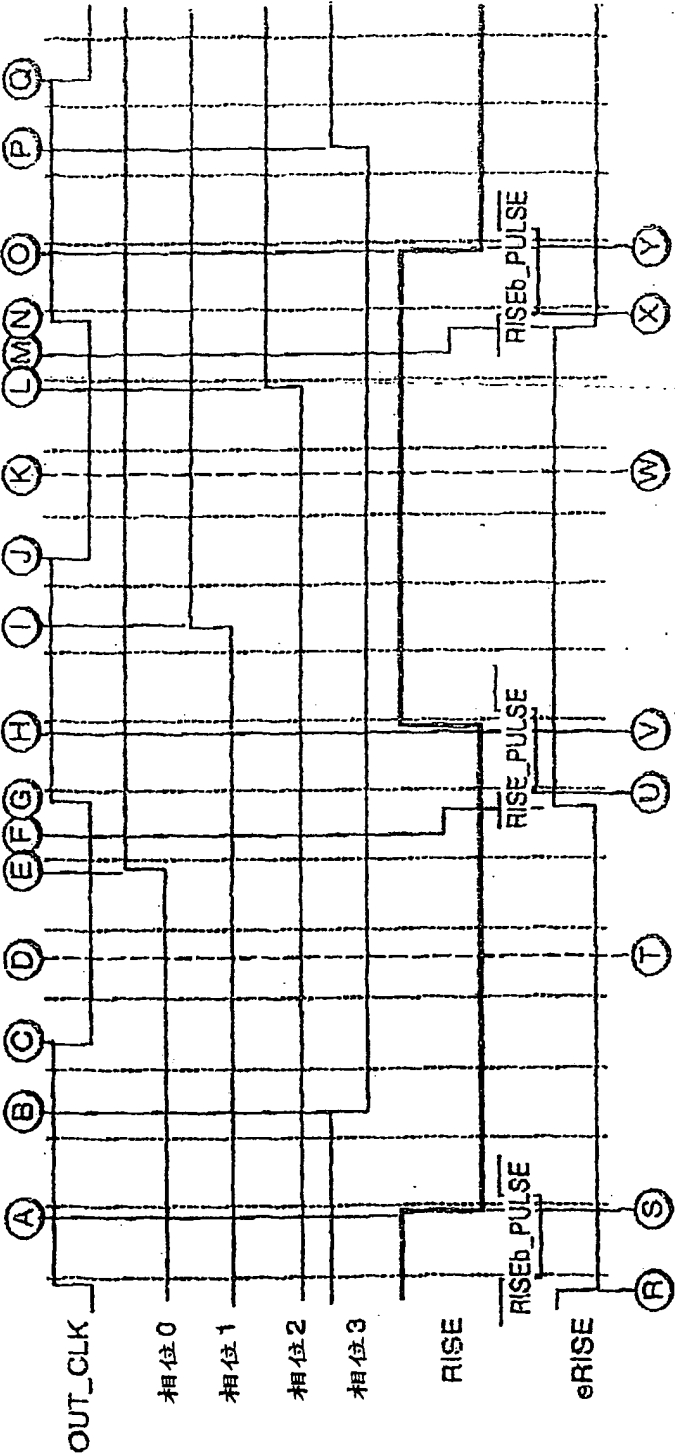


图16C

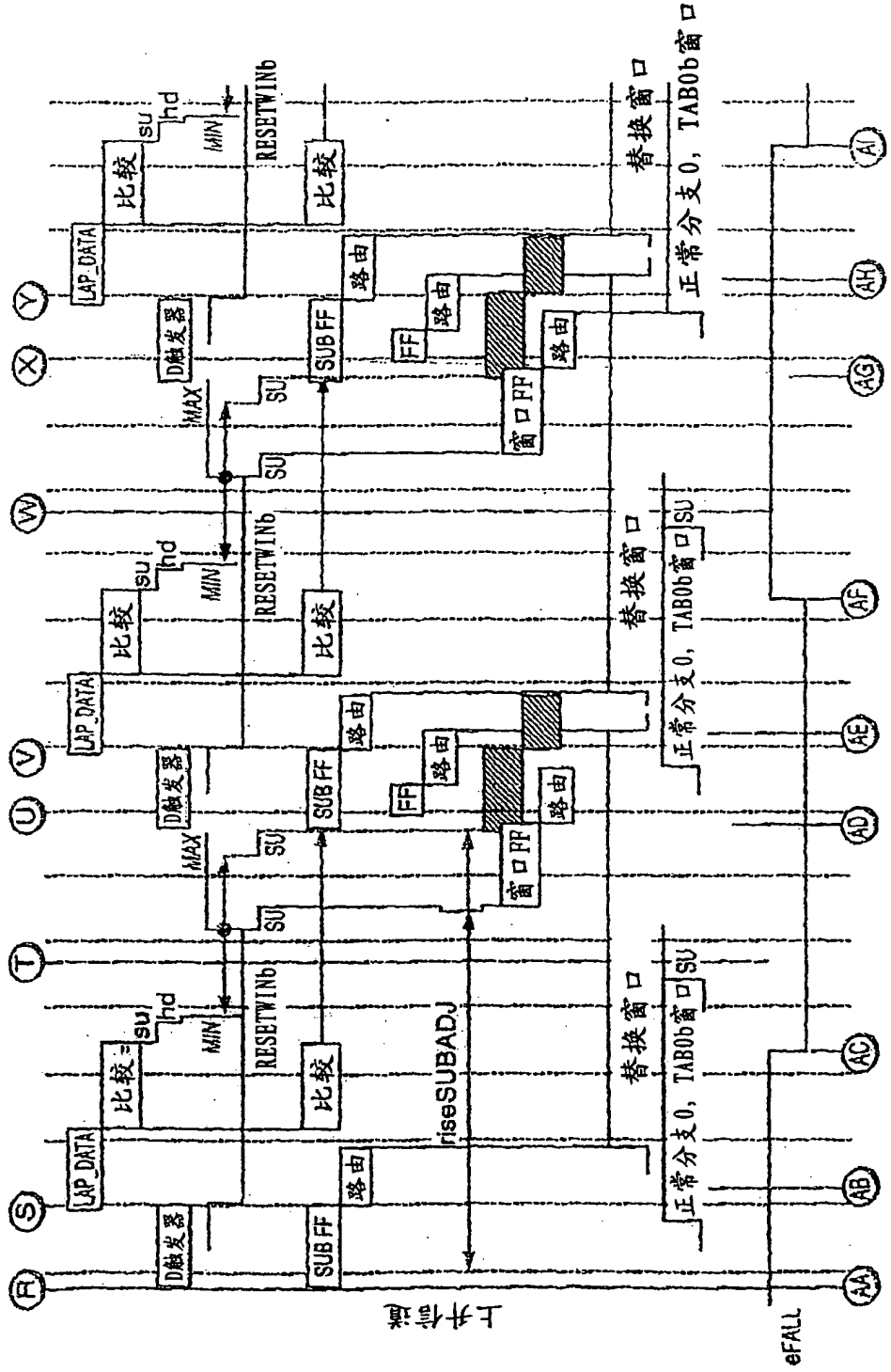


图 16D

